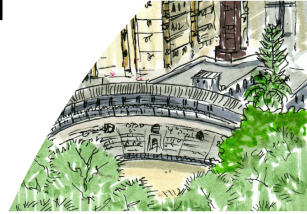


Proceedings of the

16th International Society for Music Information Retrieval Conference

October 26 - 30, 2015
Málaga, Spain



Edited by
Meinard Müller and Frans Wiering

ISMIR 2015

**Proceedings of the 16th International Society
for Music Information Retrieval Conference**



**October 26 - 30, 2015
Málaga, Spain**

**Edited by
Meinard Müller and Frans Wiering**

ISMIR 2015 is organized by the International Society for Music Information Retrieval and the ATIC Research Group of the Universidad de Málaga.

Website: <http://ismir2015.ismir.net>

<http://ismir2015.uma.es/>

Cover design by Alberto Peinado & Isabel Barbancho

Cover Málaga drawings by Cristina Urdiales

Cover G-clef drawing by Alberto Peinado & Isabel Barbancho

ISMIR 2015 logo by Alberto Peinado & Emilio Molina

Edited by:

Meinard Müller (International Audio Laboratories Erlangen)

Frans Wiering (Utrecht University)

ISBN 987-84-606-8853-2

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.

© 2015 International Society for Music Information Retrieval

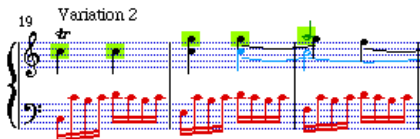
ORGANIZATION



UNIVERSIDAD
DE MÁLAGA



Escuela Técnica Superior
de Ingeniería de
Telecomunicación



SPONSORS

Gold Sponsors



UNIVERSIDAD
DE MÁLAGA



Vicerrectorado de Investigación y Transferencia

Silver Sponsors



Bronze Sponsors



Other Collaborators



Conference Committee

General Chairs

Isabel Barbancho (Universidad de Málaga, Spain)

Lorenzo J. Tardón (Universidad de Málaga, Spain)

Program Chairs

Meinard Müller (International Audio Laboratories Erlangen, Germany)

Frans Wiering (Utrecht University, The Netherlands)

Tutorial Chair

George Tzanetakis (University of Victoria, Canada)

Unconference Chairs

Dan Ellis (Columbia University, USA)

Douglas Eck (Google)

Late-Breaking and Demo Chairs

Christian Dittmar (International Audio Laboratories Erlangen, Germany)

Ana M. Barbancho (Universidad de Málaga, Spain)

Music Chairs

José Pedro Rodrigues Magalhães (Chordify, The Netherlands)

Carlos Guedes (New York University, Abu Dhabi)

Music Curator

Robert Rowe (New York University, USA)

Jam Session Chairs

Oriol Nieto (Pandora, USA)

Emilio Molina (Universidad de Málaga, Spain)

Local Arrangement Chairs

Alberto Peinado (Universidad de Málaga, Spain)

Andrés Ortiz (Universidad de Málaga, Spain)

Jorge Munilla (Universidad de Málaga, Spain)

José L. Santacruz (Universidad de Málaga, Spain)

Jesús Corral (Universidad de Málaga, Spain)

Hande Başak (Anadolu University, Turkey)

José Manuel Peula Palacios (Universidad de Málaga, Spain)

Manuel Fernández Carmona (Universidad de Málaga, Spain)

Joaquín Ballesteros (Universidad de Málaga, Spain)

Program Committee

Amélie Anglade, Freelance MIR and Data Science Consultant, Germany
Jean-Julien Aucouturier, CNRS/IRCAM, France
Eric Battenberg, Gracenote, USA
Juan Pablo Bello, New York University, USA
Emmanouil Benetos, Queen Mary University of London, UK
Dawn Black, Queen Mary University of London, UK
Sebastian Böck, Johannes Kepler University, Austria
Ashley Burgoyne, University of Amsterdam, The Netherlands
Ching-Hua Chuan, University of North Florida, USA
Tom Collins, De Montfort University, UK
Sally Jo Cunningham, The University of Waikato, New Zealand
Roger Dannenberg, Carnegie Mellon University, USA
Matthew Davies, INESC TEC, Portugal
Johanna Devaney, The Ohio State University, USA
Christian Dittmar, AudioLabs, Germany
Simon Dixon, Queen Mary University of London, UK
J. Stephen Downie, University of Illinois at Urbana-Champaign, USA
Zhiyao Duan, University of Rochester, USA
Andreas Ehmann, Pandora, USA
Dan Ellis, Columbia University, USA
Sebastian Ewert, Queen Mary University of London, UK
George Fazekas, Queen Mary University of London, UK
Arthur Flexer, Austrian Research Institute for Artificial Intelligence (OFAI), Austria
José Fornari, NICS - UNICAMP, Brasil
Ichiro Fujinaga, McGill University, Canada
Emilia Gómez, Universitat Pompeu Fabra, Spain
Masataka Goto, AIST, Japan
Fabien Gouyon, Pandora, USA
Maarten Grachten, OFAI, Austria
Bas de Haas, Utrecht University, The Netherlands
Dorien Herremans, Universiteit Antwerpen, Belgium
Perfecto Herrera, Universitat Pompeu Fabra, Spain
Andre Holzapfel, Bogazici University, Turkey
Xiao Hu, University of Hong Kong, Hong Kong
Eric Humphrey, MuseAmi/NYU, USA
Ozgur Izmirli, Connecticut College, USA
Peter Knees, Johannes Kepler University Linz, Austria

Peter van Kranenburg, Meertens Institute, Amsterdam, The Netherlands
Audrey Laplante, Université de Montréal, Canada
Jinha Lee, University of Washington, USA
Cynthia Liem, Delft University of Technology, The Netherlands
Matthias Mauch, Queen Mary University of London, UK
Matt McVicar, The University of Bristol, UK
Nicola Orio, University of Padua, Italy
Geoffroy Peeters, UMR IRCAM CNRS STMS, France
Marcelo Queiroz, University of Sao Paulo, Brasil
Christophe Rhodes, Goldsmiths, University of London, UK
Justin Salamon, New York University, USA
Erik Schmidt, Pandora, USA
Jeffrey Scott, Gracenote, USA
Xavier Serra, Universitat Pompeu Fabra, Spain
Jordan Smith, AIST, Japan
Mohamed Sordo, University of Miami, USA
Bob Sturm, Aalborg University, Denmark
Li Su, Academia Sinica, Taiwan
Douglas Turnbull, Ithaca College, USA
George Tzanetakis, University of Victoria, Canada
Julian Urbano, Universitat Pompeu Fabra, Spain
Ju-Chiang Wang, Academia Sinica, Taiwan
Yi-Hsuan Yang, Academia Sinica, Taiwan

Reviewers

Abesser, Jakob	Chen, Chih-Ming	Giraud, Mathieu
Ahonen, Teppo	Cherla, Srikanth	Goebel, Werner
Albrecht, Joshua	Chin, Yu-Hao	Gomez, Daniel
Aljanaki, Anna	Choi, Kahyun	Grill, Thomas
Anden, Joakim	Chung, Chia-Hao	Grohgan, Harald
Arjannikov, Tom	Cogliati, Andrea	Grunberg, David
Arzt, Andreas	Collins, Nick	Guastavino, Catherine
	Cont, Arshia	Gulati, Sankalp
Bailes, Freya	Cotton, Courtenay	Gurevich, Michael
Bainbridge, David	Coviello, Emanuele	
Balke, Stefan	Crawford, Tim	Hahn, Henrik
Barbancho, Isabel		Hamanaka, Masatoshi
Barbancho, Ana Maria	David, Bertrand	Hamel, Philippe
Basaran, Dogaç	De Man, Brecht	Hankinson, Andrew
Baumann, Stephan	Degara, Norberto	Hanna, Pierre
Bay, Mert	Dessein, Arnaud	Hargreaves, Steven
Bazzica, Alessio	Di Giorgi, Bruno	Hedges, Tom
Bellogin, Alejandro	Driedger, Jonathan	Herrera, Jorge
Bemman, Brian	Durand, Simon	Hoover, Amy
Bennett, Christopher		Huang, Anna
Bernardes, Gilberto	Eghbal-Zadeh, Hamid	Huang, Po-Sen
Bigo, Louis	Elowsson, Anders	
Bimbot, Frédéric	Essinger, Steve	Imbrasaite, Vaiva
Bittner, Rachel		Ishwar, Vignesh
Bohak, Ciril	Faraldo, Ángel	Itoyama, Katsutoshi
Bonafonte, Antonio	Fernandes Tavares, Tiago	
Bosch, Juanjo	Fields, Ben	Jakubowski, Kelly
Bountouridis, Dimitrios	Fillon, Thomas	Janssen, Berit
	Fitzgerald, Derry	Jao, Ping-Keng
Caetano, Marcelo	Fonseca, Nuno	Jensen, Kristoffer
Cambouropoulos, Emilios	Font, Frederic	Jordá, Sergi
Cancino Chacon, Carlos E.	Foster, Peter	
Cano, Estefania	Fukayama, Satoru	Kaneshiro, Blair
Carabias, Julio		Keller, Damián
Cartwright, Mark	Ganseman, Joachim	Kelz, Rainer
Chan, Tak-Shing	Gedik, Ali Cenk	Kim, Minje
Chen, Shuo	Gingras, Bruno	Klapuri, Anssi

Knopke, Ian	McFee, Brian	Quinton, Elio
Koerich, Alessandro	McKay, Cory	
Koops, Vincent	Meseguer, Gabriel	Raczynski, Stanislaw
Korzeniowski, Filip	Mesnage, Cedric	Raffel, Colin
Kosta, Katerina	Milne, Andrew	Rafii, Zafar
Krebs, Florian	Miron, Marius	Ramirez, Miguel
Kroher, Nadine	Moore, Josh	Rao, Preeti
Kruspe, Anna	Morvidone, Marcela	Rashkov, Valery
Ku, Lun-Wei	Moussallam, Manuel	Rauber, Andreas
	Müllensiefen, Daniel	Rebelo, Ana
Lagrange, Mathieu		Ren, Gang
Langlois, Thibault	Nagano, Hidehisa	Ringwalt, Daniel
Lartillot, Olivier	Nakano, Tomoyasu	Rizo, David
Lattner, Stefan	Nakano, Masahiro	Robertson, Andrew
Lee, Kyogu	Nam, Juhan	Robine, Matthias
Lemstrom, Kjell	Naveda, Luiz	Rocamora, Martin
Leve, Florence	Neubarth, Kerstin	Rocha, Bruno
Levy, Mark	Nichols, Eric	Rodriguez Algarra, Francisco
Lewis, David	Nieto, Oriol	Rodríguez López, Marcelo
Lewis, Richard		Roma, Gerard
Li, Bochen	Ogihara, Mitsunori	Rowe, Robert
Li, Wei	Oland, Anders	Russo, Frank
Liang, Dawen	Oramas, Sergio	
Lidy, Thomas	Ostuni, Vito	
Lin, Yin-Tzu		Saari, Pasi
Liu, Yi-Wen	Page, Kevin	Sadakata, Makiko
Liu, Jen-Yu	Panteli, Maria	Said, Alan
Liutkus, Antoine	Papadopoulos, Héléne	Schindler, Alexander
Loviscach, Jörn	Paulus, Jouni	Schlüter, Jan
Lu, Lie	Pauwels, Johan	Schuller, Björn
Lukashevich, Hanna	Percival, Graham	Seetharaman, Prem
	Perez, Alfonso	Sentürk, Sertan
Machado, Anderson	Pertusa, Antonio	Sigler, Andie
Magalhães, José Pedro	Pesek, Matevž	Sigtia, Siddharth Sanjay
Mandel, Michael	Pikrakis, Aggelos	Smaragdis, Paris
Margulis, Elizabeth	Prätzlich, Thomas	Smith, Lloyd
Marolt, Matija	Preinfalk, Martin	Soleymani, Mohammad
Marrero, Mónica	Prockup, Matthew	Song, Yading
Marsden, Alan	Proutskova, Polina	Sonnleitner, Reinhard
Martorell, Agustin	Pugin, Laurent	Souza Britto Jr., Alceu de
Mayer, Rudolf		Sprechmann, Pablo

Srinivasamurthy, Ajay	Valk, Reinier de	Weij, Bastiaan van der
Stewart, Rebecca	Vall, Andreu	Weiss, Ron
Stöter, Fabian-Robert	Valle, Rafael	Weiß, Christof
Stowell, Dan	Van Balen, Jan	Whalley, Ian
Subramanian, Anand	Van Handel, Leigh	Wolff, Daniel
Summers, Cameron	Velarde, Gissel	Wu, Fu-Hai Frank
Supper, Ben	Veltkamp, Remco	Wu, Ben
	Vempala, Naresh	
Temperley, David	Vigliensoni, Gabriel	Xia, Guangyu
Thalmann, Florian	Vogl, Richard	
Tian, Mi		Zacharakis, Asteris
Tjoa, Steve	Wang, Xing	Zanoni, Massimiliano
Tkalcic, Marko	Wang, Siying	Zbikowski, Lawrence
Toussaint, Godfried	Wang, Xinxi	
	Wang, Hsin-Min	
Uhle, Christian	Weerkamp, Wouter	
Upham, Finn	Weigl, David	

Preface

It is our great pleasure to welcome you to the 16th International Society for Music Information Retrieval Conference (ISMIR 2015). The annual ISMIR conference is the world's leading research forum on processing, analyzing, searching, organizing, and accessing music-related data. This year's conference, which takes place in Málaga, Spain, from October 26th – 30th, 2015, is organized by the ATIC Research Group of the Universidad de Málaga.

The present volume contains the complete manuscripts of all the peer-reviewed papers presented at ISMIR 2015. A total of 278 submissions were received before the deadline, out of which 242 complete and well-formatted papers entered the review process. Special care was taken to assemble an experienced and interdisciplinary review panel including people from many different institutions worldwide. As in previous years, the reviews were double-blinded (i.e., both the authors and the reviewers were anonymous) with a two-tier review model involving a pool of 257 reviewers and a program committee. Reviewers and PC members were able to bid for papers. Each paper was assigned to a PC member and three reviewers. Reviewer assignments were based on topic preferences, bids, and PC member assignments. After the review phase, PC members and reviewers entered a (name-disclosed) discussion phase aiming to homogenize acceptance vs. rejection decisions.

Compared to previous years, the size of the program committee increased significantly and now comprises 61 members. Taking care of four submissions on average, the PC members were asked to adopt an active role in the review process by conducting an intensive discussion phase with the other reviewers and by providing a detailed meta-review. Final acceptance decisions were based on 973 reviews and meta-reviews. From the 242 reviewed papers, 114 papers were accepted resulting in an acceptance rate of 47.1%. The table shown on the next page summarizes the ISMIR publication statistics over the last years.

The mode of presentation of the papers was determined after the accept/reject decisions and has no relation to the quality of the papers or to the number of pages allotted in the proceedings. From the 114 contributions, 24 papers were chosen for oral presentation based on the topic and broad appeal of the work, whereas the other 90 were chosen for poster presentation. Oral presentations have a 20-minute slot (including setup and questions/answers from the audience) whereas poster presentations are done in two sessions per day, the same posters being presented in the morning and in the afternoon of a given conference day.

Year	Location	Oral	Poster	Total Papers	Total Pages	Total Authors	Unique Authors	Pages/ Paper	Authors/ Paper	Unique Authors/ Paper
2000	Plymouth	19	16	35	155	68	63	4.4	1.9	1.8
2001	Indiana	25	16	41	222	100	86	5.4	2.4	2.1
2002	Paris	35	22	57	300	129	117	5.3	2.3	2.1
2003	Baltimore	26	24	50	209	132	111	4.2	2.6	2.2
2004	Barcelona	61	44	105	582	252	214	5.5	2.4	2
2005	London	57	57	114	697	316	233	6.1	2.8	2
2006	Victoria	59	36	95	397	246	198	4.2	2.6	2.1
2007	Vienna	62	65	127	486	361	267	3.8	2.8	2.1
2008	Philadelphia	24	105	105	630	296	253	6	2.8	2.4
2009	Kobe	38	85	123	729	375	292	5.9	3	2.4
2010	Utrecht	24	86	110	656	314	263	6	2.	2.4
2011	Miami	36	97	133	792	395	322	6	3	2.4
2012	Porto	36	65	101	606	324	264	6	3.2	2.6
2013	Curitiba	31	67	98	587	395	236	5.9	3	2.4
2014	Taipei	33	73	106	635	343	271	6	3.2	2.6
2015	Málaga	24	90	114	792	370	296	7	3.2	2.6

The ISMIR 2015 conference runs for a 5-day period. The selected submissions are presented over a period of 3.5 days, preceded by a day of tutorials and followed by half a day of late-breaking/demo & unconference sessions. Moreover, a satellite event called “Hacking Audio and Music Research” (HAMR) is offered, which is held on October 24th and 25th. We now give a summary of the highlights of the conference.

Tutorials

Six tutorials take place on Monday, providing a good balance between culture and technology. Three 3-hour tutorials are presented in parallel on Monday morning, and three in parallel on Monday afternoon.

Morning sessions:

- Tutorial 1: Why singing is interesting (Simon Dixon, Masataka Goto, Matthias Mauch)
- Tutorial 2: Addressing the music information needs of musicologists (Richard J. Lewis, Ben Fields, Tim Crawford)
- Tutorial 3: Markov logic networks for music analysis (Helene Papadopoulos)

Afternoon sessions:

- Tutorial 4: COMputation and FLAmenco: Why flamenco is interesting for MIR research (Emilia Gómez, Nadine Kroher, Jose Miguel Díaz-Báñez, Sergio Oramas, Joaquín Mora, Francisco Gómez-Martín)
- Tutorial 5: Using correlation analysis and big data to identify and predict musical behaviors (Jeff C. Smith)
- Tutorial 6: Automatic music transcription (Zhiyao Duan, Emmanouil Benetos)

Keynote Speakers

We are honored to have two distinguished keynote speakers:

- Prof. Mark Sandler from Queen Mary University of London, UK.
- Prof. J. Stephen Downie from the University of Illinois at Urbana-Champaign, USA.

MIREX

The Music Information Retrieval Evaluation eXchange (MIREX) is a collective effort to evaluate cutting-edge methods for various MIR tasks. Since 2005, MIREX has been an integral part of ISMIR. This year, ISMIR features the following MIREX-related events:

- Plenary summary of tasks and results of MIREX 2015
- Report of the “Grand Challenge 2015: User Experience” (GC15UX)
- Plenary discussion of MIREX 2016 and GC16UX
- Poster session of MIREX 2015 participants

Late-Breaking/Demo & Unconference

Friday afternoon is dedicated to late-breaking papers and MIR system demonstrations. Abstracts for these presentations are available online. Moreover, as in previous years, we have a special “unconference” session in which participants break up into smaller groups to discuss MIR issues of particular interest. This is an informal and informative opportunity to get to know peers and colleagues from all around the world.

Music Program

ISMIR 2015 includes a music program, which is centered around one curated concert that takes place on Wednesday, October 24th, at Sala Unicaja de Conciertos María Cristina. The aim of the concert program is two-fold: to encourage the use of Music Information Retrieval (MIR) techniques in the creation of new music and to explore music that can suggest novel ideas for research in the MIR field.

Social Events

ISMIR 2015 not only offers interesting papers, posters, and tutorials, but also aims at giving the participants an unforgettable stay. The social program provides participants with an opportunity to relax after meetings, to experience Málaga, and to network with other ISMIR participants. The social program includes:

- Monday, October 26, Welcome cocktail at “Vinoteca Museo Los Patios de Beatas”.
- Wednesday, October 28, Concert at “Sala de Conciertos María Cristina” that includes, not only the music from the “ISMIR 2015 Call for Music” but also Live Flamenco Dancing & Music.
- Thursday, October 29, Gala Dinner at “Hacienda del Alamo” and ISMIR 2015 Pandora Jam Session.

Conference Venue

More than 3,000 years of history have passed since Málaga’s establishment by the Phoenicians. Today, Málaga is a beautiful, friendly, and cosmopolitan city that enchants tourists from all over the world. The conference venue is the hotel NH MALAGA, located in the heart of Málaga’s city centre. The venue is within walking distance to main attractions, including the Cathedral, the Picasso Museum, Larios Street, Muelle 1 (Pier 1), Alcazaba, and the sea promenade. Be sure to take the chance to explore Málaga and enjoy your stay in Spain!

Acknowledgments

We are very proud to present to you the proceedings of ISMIR 2015. The conference program was made possible thanks to the hard work of many people including the members of the local organization team, the many reviewers, the meta-reviewers from the program committee, and the members of the conference committee. Special thanks go to this year's sponsors:

- Gold Sponsors:
 - Fundación Española para la Ciencia y la Tecnología (FECYT), Ministerio de Economía y Competitividad, Gobierno de España (under project FCT-14-8217)
 - Shazam
 - Gracenote
 - Pandora
 - Bose Corporation
 - Universidad de Málaga, Vicerrectorado de Investigación y Transferencia, Campus de Excelencia Internacional Andalucía Tech

- Silver Sponsors:
 - Steinberg Media Technologies GmbH
 - Native Instruments GmbH

- Bronze Sponsors:
 - Google, Inc
 - ACRCLOUD - Automatic Content Recognition Cloud Service
 - Smule

- Other Collaborators:
 - Departamento de Ingeniería de Comunicaciones, Universidad de Málaga
 - E.T.S. Ingeniería de Telecomunicación, Universidad de Málaga
 - Fundación Unicaja
 - Ayuntamiento de Málaga
 - Málaga Convention Bureau

Last, but not least, the ISMIR program is possible only thanks to the excellent contributions of our community in response to our call for participation. The biggest acknowledgment goes to you, the authors, researchers and participants of this conference. We hope you all have a wonderful and unforgettable stay in Spain!

Isabel Barbancho
Lorenzo J. Tardón
General Chairs, ISMIR 2015

Meinard Müller
Frans Wiering
Program Chairs, ISMIR 2015

Contents

Keynote Talks	1
Integrating Music Information Sources for Music Production and Consumption	
<i>Mark Sandler</i>	3
The Promise of Music Information Retrieval: Are we there yet?	
<i>J. Stephen Downie</i>	5
Tutorials	7
Why Singing is Interesting	
<i>Simon Dixon, Masataka Goto, Matthias Mauch</i>	9
Addressing the Music Information Needs of Musicologists	
<i>Richard J. Lewis, Ben Fields, Tim Crawford</i>	10
Markov Logic Networks for Music Analysis	
<i>Helene Papadopoulou</i>	11
COMputation and FLAmenco: Why Flamenco is Interesting for MIR Research	
<i>Emilia Gómez, Nadine Kroher, Jose Miguel Díaz-Báñez, Sergio Oramas, Joaquín Mora, Francisco Gómez-Martín</i>	12
Using Correlation Analysis and Big Data to Identify and Predict Musical Behaviors	
<i>Jeff C. Smith</i>	13
Automatic Music Transcription	
<i>Zhiyao Duan, Emmanouil Benetos</i>	14
Poster Session 1	15
Image Quality Estimation for Multi-Score OMR	
<i>Dan Ringwalt, Roger B. Dannenberg</i>	17
Comparative Music Similarity Modelling Using Transfer Learning Across User Groups	
<i>Daniel Wolff, Andrew MacFarlane, Tillman Weyde</i>	24
Modeling Genre with the Music Genome Project: Comparing Human-Labeled Attributes and Audio Features	
<i>Matthew Prockup, Andreas F. Ehmman, Fabien Gouyon, Erik M. Schmidt, Oscar Celma, Youngmoo E. Kim</i>	31
Cover Song Identification with Timbral Shape Sequences	
<i>Christopher J. Tralie, Paul Bendich</i>	38
On the Impact of Key Detection Performance for Identifying Classical Music Styles	
<i>Christof Weiß, Maximilian Schaab</i>	45
Chord Detection Using Deep Learning	
<i>Xinquan Zhou, Alexander Lerch</i>	52
Temporal Music Context Identification with User Listening Data	
<i>Cameron Summers, Phillip Popp</i>	59
Improving Music Recommendations with a Weighted Factorization of the Tagging Activity	
<i>Andreu Vall, Marcin Skowron, Peter Knees, Markus Schedl</i>	65

An Efficient State-Space Model for Joint Tempo and Meter Tracking <i>Florian Krebs, Sebastian Böck, Gerhard Widmer</i>	72
Automatic Handwritten Mensural Notation Interpreter: From Manuscript to MIDI Performance <i>Yu-Hui Huang, Xuanli Chen, Serafina Beck, David Burn, Luc Van Gool</i>	79
Infinite Superimposed Discrete All-Pole Modeling for Multipitch Analysis of Wavelet Spectrograms <i>Kazuyoshi Yoshii, Katsutoshi Itoyama, Masataka Goto</i>	86
Melodic Similarity in Traditional French-Canadian Instrumental Dance Tunes <i>Laura Risk, Lillio Mok, Andrew Hankinson, Julie Cumming</i>	93
A Semantic-Based Approach for Artist Similarity <i>Sergio Oramas, Mohamed Sordo, Luis Espinosa-Anke, Xavier Serra</i>	100
Predicting Pairwise Pitch Contour Relations Based on Linguistic Tone Information in Beijing Opera Singing <i>Shuo Zhang, Rafael Caro Repetto, Xavier Serra</i>	107
Song2Quartet: A System for Generating String Quartet Cover Songs from Polyphonic Audio of Popular Music <i>Graham Percival, Satoru Fukayama, Masataka Goto</i>	114
Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks <i>Jan Schlüter, Thomas Grill</i>	121
Audio Chord Recognition with a Hybrid Recurrent Neural Network <i>Siddharth Sigtia, Nicolas Boulanger-Lewandowski, Simon Dixon</i>	127
Design and Evaluation of a Probabilistic Music Projection Interface <i>Beatrix Vad, Daniel Boland, John Williamson, Roderick Murray-Smith, Peter Berg Steffensen</i>	134
Conceptual Blending in Music Cadences: A Formal Model and Subjective Evaluation <i>Asterios Zacharakis, Maximos Kaliakatsos-Papakostas, Emilios Cambouropoulos</i>	141
Harmonic-Percussive Source Separation Using Harmonicity and Sparsity Constraints <i>Jeongsoo Park, Kyogu Lee</i>	148
A Hierarchical Bayesian Framework for Score-Informed Source Separation of Piano Music Signals <i>Wai Man Szeto, Kin Hong Wong</i>	155
Automatic Tune Family Identification by Musical Sequence Alignment <i>Patrick E. Savage, Quentin D. Atkinson</i>	162
Evaluation of Album Effect for Feature Selection in Music Genre Recognition <i>Igor Votolkin, Günter Rudolph, Claus Weihs</i>	169
Music Pattern Discovery with Variable Markov Oracle: A Unified Approach to Symbolic and Audio Representations <i>Cheng-i Wang, Jennifer Hsu, Shlomo Dubnov</i>	176
Automatic Solfège Assessment <i>Rodrigo Schramm, Helena de Souza Nunes, Cláudio Rosito Jung</i>	183

Evaluating Conflict Management Mechanisms for Online Social Jukeboxes <i>Felipe Vieira, Nazareno Andrade</i>	190
Particle Filters for Efficient Meter Tracking with Dynamic Bayesian Networks <i>Ajay Srinivasamurthy, Andre Holzapfel, Ali Taylan Cemgil, Xavier Serra</i>	197
Analysis of the Evolution of Research Groups and Topics in the ISMIR Conference <i>Mohamed Sordo, Mitsunori Ogihara, Stefan Wuchty</i>	204
A Toolkit for Live Annotation of Opera Performance: Experiences Capturing Wagner’s Ring Cycle <i>Kevin R. Page, Terhi Nurmikko-Fuller, Carolin Rindfleisch, David M. Weigl, Richard Lewis, Laurence Dreyfus, David De Roure</i>	211
Selective Acquisition Techniques for Enculturation-Based Melodic Phrase Segmentation <i>Marcelo E. Rodríguez-López, Anja Volk</i>	218
Oral Session 1: Corpus Analysis & Annotation	225
Corpus Analysis Tools for Computational Hook Discovery <i>Jan Van Balen, John Ashley Burgoyne, Dimitrios Bountouridis, Daniel Müllensiefen, Remco C. Veltkamp</i>	227
Large-Scale Content-Based Matching of MIDI and Audio Files <i>Colin Raffel, Daniel P. W. Ellis</i>	234
Improving Genre Annotations for the Million Song Dataset <i>Hendrik Schreiber</i>	241
A Software Framework for Musical Data Augmentation <i>Brian McFee, Eric J. Humphrey, Juan P. Bello</i>	248
Oral Session 2: Rhythm & Beat	255
Drum Transcription Using Partially Fixed Non-Negative Matrix Factorization with Template Adaptation <i>Chih-Wei Wu, Alexander Lerch</i>	257
Beat and Downbeat Tracking Based on Rhythmic Patterns Applied to the Uruguayan Candombe Drumming <i>Leonardo Nunes, Martín Rocamora, Luis Jure, Luiz W. P. Biscainho</i>	264
Automated Estimation of Ride Cymbal Swing Ratios in Jazz Recordings <i>Christian Dittmar, Martin Pfeiderer, Meinard Müller</i>	271
Poster Session 2	279
Musical Offset Detection of Pitched Instruments: The Case of Violin <i>Che-Yuan Liang, Li Su, Yi-Hsuan Yang, Hsin-Ming Lin</i>	281
Specter: Combining Music Information Retrieval with Sound Spatialization <i>Bill Manaris, Seth Stoudenmier</i>	288
Content-Aware Collaborative Music Recommendation Using Pre-trained Neural Networks <i>Dawen Liang, Minshu Zhan, Daniel P. W. Ellis</i>	295

Comparative Analysis of Orchestral Performance Recordings: An Image-Based Approach <i>Cynthia C. S. Liem, Alan Hanjalic</i>	302
Monaural Blind Source Separation in the Context of Vocal Detection <i>Bernhard Lehner, Gerhard Widmer</i>	309
Detection of Common Mistakes in Novice Violin Playing <i>Yin-Jyun Luo, Li Su, Yi-Hsuan Yang, Tai-Shih Chi</i>	316
Probabilistic Modular Bass Voice Leading in Melodic Harmonisation <i>Dimos Makris, Maximos Kaliakatsos-Papakostas, Emiliios Cambouropoulos</i>	323
An Iterative Multi Range Non-Negative Matrix Factorization Algorithm for Polyphonic Music Transcription <i>Anis Khlif, Vidhyasaharan Sethu</i>	330
Training Phoneme Models for Singing with “Songified” Speech Data <i>Anna M. Kruspe</i>	336
Graph-Based Rhythm Interpretation <i>Rong Jin, Christopher Raphael</i>	343
Let it Bee – Towards NMF-Inspired Audio Mosaicing <i>Jonathan Driedger, Thomas Prätzlich, Meinard Müller</i>	350
Real-Time Music Tracking Using Multiple Performances as a Reference <i>Andreas Arzt, Gerhard Widmer</i>	357
Two Data Sets for Tempo Estimation and Key Detection in Electronic Dance Music Annotated from User Corrections <i>Peter Knees, Ángel Faraldo, Perfecto Herrera, Richard Vogl, Sebastian Böck, Florian Hörschläger, Mickael Le Goff</i>	364
Towards Support for Understanding Classical Music: Alignment of Content Descriptions on the Web <i>Taku Kuribayashi, Yasuhito Asano, Masatoshi Yoshikawa</i>	371
FlaBase: Towards the Creation of a Flamenco Music Knowledge Base <i>Sergio Oramas, Francisco Gómez, Emilia Gómez, Joaquín Mora</i>	378
Discovery of Syllabic Percussion Patterns in Tabla Solo Recordings <i>Swapnil Gupta, Ajay Srinivasamurthy, Manoj Kumar, Hema A. Murthy, Xavier Serra</i>	385
Autoregressive Hidden Semi-Markov Model of Symbolic Music Performance for Score Following <i>Eita Nakamura, Philippe Cuvillier, Arshia Cont, Nobutaka Ono, Shigeki Sagayama</i>	392
Automatic Mashup Creation by Considering both Vertical and Horizontal Mashabilities <i>Chuan-Lung Lee, Yin-Tzu Lin, Zun-Ren Yao, Feng-Yi Lee, Ja-Ling Wu</i>	399
Hierarchical Evaluation of Segment Boundary Detection <i>Brian McFee, Oriol Nieto, Juan P. Bello</i>	406
Improving MIDI Guitar’s Accuracy with NMF and Neural Net <i>Masaki Otsuka, Tetsuro Kitahara</i>	413
Analysis of Intonation Trajectories in Solo Singing <i>Jiajie Dai, Matthias Mauch, Simon Dixon</i>	420

Evaluating the General Chord Type Representation in Tonal Music and Organising GCT Chord Labels in Functional Chord Categories	
<i>Maximos Kaliakatos-Papakostas, Asterios Zacharakis, Costas Tsougras, Emilios Cambouropoulos</i>	427
Beat Histogram Features from NMF-Based Novelty Functions for Music Classification	
<i>Athanasios Lykartsis, Chih-Wei Wu, Alexander Lerch</i>	434
Music Shapelets for Fast Cover Song Recognition	
<i>Diego F. Silva, Vinícius M. A. Souza, Gustavo E. A. P. A. Batista</i>	441
Improving Score-Informed Source Separation for Classical Music through Note Refinement	
<i>Marius Miron, Julio José Carabias-Orti, Jordi Janer</i>	448
In Their Own Words: Using Text Analysis to Identify Musicologists' Attitudes towards Technology	
<i>Charles Inskip, Frans Wiering</i>	455
Combining Features for Cover Song Identification	
<i>Julien Osmalskiy, Peter Foster, Simon Dixon, Jean-Jacques Embrechts</i>	462
Score Following for Piano Performances with Sustain-Pedal Effects	
<i>Bochen Li, Zhiyao Duan</i>	469
Understanding Users of Commercial Music Services through Personas: Design Implications	
<i>Jin Ha Lee, Rachel Price</i>	476
Corpus-Based Rhythmic Pattern Analysis of Ragtime Syncopation	
<i>Hendrik Vincent Koops, Anja Volk, W. Bas de Haas</i>	483
Oral Session 3: Melody & Voice	491
Comparing Voice and Stream Segmentation Algorithms	
<i>Nicolas Guiomard-Kagan, Mathieu Giraud, Richard Groult, Florence Levé</i>	493
Melody Extraction by Contour Classification	
<i>Rachel M. Bittner, Justin Salamon, Slim Essid, Juan P. Bello</i>	500
Comparison of the Singing Style of Two Jingju Schools	
<i>Rafael Caro Repetto, Rong Gong, Nadine Kroher, Xavier Serra</i>	507
Oral Session 4: Mixed	515
Improving Optical Music Recognition by Combining Outputs from Multiple Sources	
<i>Victor Padilla, Alex McLean, Alan Marsden, Kia Ng</i>	517
Relating Natural Language Text to Musical Passages	
<i>Richard Sutcliffe, Tim Crawford, Chris Fox, Deane L. Root, Eduard Hovy, Richard Lewis</i>	524
Music Boundary Detection Using Neural Networks on Combined Features and Two-Level Annotations	
<i>Thomas Grill, Jan Schlüter</i>	531
Neuroimaging Methods for Music Information Retrieval: Current Findings and Future Prospects	
<i>Blair Kaneshiro, Jacek P. Dmochowski</i>	538
Oral Session 5: Similarity	545

Improving Visualization of High-Dimensional Music Similarity Spaces	
<i>Arthur Flexer</i>	547
I-Vectors for Timbre-Based Music Similarity and Music Artist Classification	
<i>Hamid Eghbal-zadeh, Bernhard Lehner, Markus Schedl, Gerhard Widmer</i>	554
Correlating Extracted and Ground-Truth Harmonic Data in Music Retrieval Tasks	
<i>Dylan Freedman, Eddie Kohler, Hans Tutschku</i>	561
Poster Session 3	569
Classical Music on the Web – User Interfaces and Data Representations	
<i>Martin Gasser, Andreas Arzt, Thassilo Gadermaier, Maarten Grachten, Gerhard Widmer</i>	571
A Statistical View on the Expressive Timing of Piano Rolled Chords	
<i>Mutian Fu, Guangyu Xia, Roger B. Dannenberg, Larry Wasserman</i>	578
Hybrid Long- and Short-Term Models of Folk Melodies	
<i>Srikanth Cherla, Son N. Tran, Tillman Weyde, Artur d’ Avila Garcez</i>	584
Efficient Melodic Query Based Audio Search for Hindustani Vocal Compositions	
<i>Kaustuv Kanti Ganguli, Abhinav Rastogi, Vedhas Pandit, Prithvi Kantan, Preeti Rao</i>	591
Modified Perceptual Linear Prediction Liftered Cepstrum (MPLPLC) Model for Pop Cover Song Recognition	
<i>Ning Chen, J. Stephen Downie, Haidong Xiao, Yu Zhu, Jie Zhu</i>	598
Raga Verification in Carnatic Music Using Longest Common Segment Set	
<i>Shrey Dutta, Krishnaraj Sekhar PV, Hema A. Murthy</i>	605
Instrument Identification in Optical Music Recognition	
<i>Yucong Jiang, Christopher Raphael</i>	612
Cross-Version Singing Voice Detection in Classical Opera Recordings	
<i>Christian Dittmar, Bernhard Lehner, Thomas Prätzlich, Meinard Müller, Gerhard Widmer</i>	618
Accurate Tempo Estimation Based on Recurrent Neural Networks and Resonating Comb Filters	
<i>Sebastian Böck, Florian Krebs, Gerhard Widmer</i>	625
Musicology of Early Music with Europeana Tools and Services	
<i>Erik Duval, Marnix van Berchum, Anja Jentsch, Gonzalo Alberto Parra Chico, Andreas Drakos</i>	632
Singing Voice Separation from Monaural Music Based on Kernel Back-Fitting Using Beta-Order Spectral Amplitude Estimation	
<i>Hye-Seung Cho, Jun-Yong Lee, Hyoung-Gook Kim</i>	639
Schematizing the Treatment of Dissonance in 16th-Century Counterpoint	
<i>Andie Sigler, Jon Wild, Eliot Handelman</i>	645
Predictive Power of Personality on Music-Genre Exclusivity	
<i>Jothi Bansal, Matthew Woolhouse</i>	652
A Comparison of Symbolic Similarity Measures for Finding Occurrences of Melodic Segments	
<i>Berit Janssen, Peter van Kranenburg, Anja Volk</i>	659

PAD and SAD: Two Awareness-Weighted Rhythmic Similarity Distances <i>Daniel Gómez-Marín, Sergi Jordà, Perfecto Herrera</i>	666
Four Timely Insights on Automatic Chord Estimation <i>Eric J. Humphrey, Juan P. Bello</i>	673
Improving Melodic Similarity in Indian Art Music Using Culture-Specific Melodic Characteristics <i>Sankalp Gulati, Joan Serrà, Xavier Serra</i>	680
Searching Lyrical Phrases in A-Capella Turkish Makam Recordings <i>Georgi Dzhambazov, Sertan Şentürk, Xavier Serra</i>	687
Quantifying Lexical Novelty in Song Lyrics <i>Robert J. Ellis, Zhe Xing, Jiakun Fang, Ye Wang</i>	694
An Efficient Temporally-Constrained Probabilistic Model for Multiple-Instrument Music Transcription <i>Emmanouil Benetos, Tillman Weyde</i>	701
Electric Guitar Playing Technique Detection in Real-World Recording Based on F0 Sequence Pattern Recognition <i>Yuan-Ping Chen, Li Su, Yi-Hsuan Yang</i>	708
Extending a Model of Monophonic Hierarchical Music Analysis to Homophony <i>Phillip B. Kirlin, David L. Thomas</i>	715
The MIR Perspective on the Evolution of Dynamics in Mainstream Music <i>Emmanuel Deruty, François Pachet</i>	722
Theme And Variation Encodings with Roman Numerals (TAVERN): A New Data Set for Symbolic Music Analysis <i>Johanna Devaney, Claire Arthur, Nathaniel Condit-Schultz, Kirsten Nisula</i>	728
Benford's Law for Music Analysis <i>Isabel Barbancho, Lorenzo J. Tardón, Ana M. Barbancho, Mateu Sbert</i>	735
An Audio to Score Alignment Framework Using Spectral Factorization and Dynamic Time Warping <i>Julio José Carabias-Orti, F. J. Rodríguez-Serrano, P. Vera-Candeas, N. Ruiz-Reyes, F. J. Cañadas-Quesada</i>	742
New Sonorities for Early Jazz Recordings Using Sound Source Separation and Automatic Mixing Tools <i>Daniel Matz, Estefanía Cano, Jakob Abeßer</i>	749
Automatic Transcription of Ornamented Irish Traditional Flute Music Using Hidden Markov Models <i>Peter Jančovič, Münevver Köküer, Wrena Baptiste</i>	756
Towards Music Imagery Information Retrieval: Introducing the OpenMIIR Dataset of EEG Recordings from Music Perception and Imagination <i>Sebastian Stober, Avital Sternin, Adrian M. Owen, Jessica A. Grahn</i>	763
Emotion Based Segmentation of Musical Audio <i>Anna Aljanaki, Frans Wiering, Remco C. Veltkamp</i>	770

Oral Session 6: User & Community	777
MIREX Grand Challenge 2014 User Experience: Qualitative Analysis of User Feedback <i>Jim Ha Lee, Xiao Hu, Kahyun Choi, J. Stephen Downie</i>	779
AcousticBrainz: A Community Platform for Gathering Music Information Obtained from Audio <i>Alastair Porter, Dmitry Bogdanov, Robert Kaye, Roman Tsukanov, Xavier Serra</i>	786
How Music Alters Decision Making - Impact of Music Stimuli on Emotional Classification <i>Elad Liebman, Peter Stone, Corey N. White</i>	793
Put the Concert Attendee in the Spotlight. A User-Centered Design and Development Approach for Classical Concert Applications <i>Mark S. Melenhorst, Cynthia C. S. Liem</i>	800
Oral Session 7: Performance	807
Analysis of Expressive Musical Terms in Violin Using Score-Informed and Expression-Based Audio Features <i>Pei-Ching Li, Li Su, Yi-Hsuan Yang, Alvin W. Y. Su</i>	809
Spectral Learning for Expressive Interactive Ensemble Music Performance <i>Guangyu Xia, Yun Wang, Roger B. Dannenberg, Geoffrey Gordon</i>	816
Score-Informed Analysis of Intonation and Pitch Modulation in Jazz Solos <i>Jakob Abeßer, Estefanía Cano, Klaus Frieler, Martin Pfeleiderer, Wolf-Georg Zaddach</i>	823
Author Index	831

Keynote Talks

Keynote Talk 1

Integrating Music Information Sources for Music Production and Consumption

Mark Sandler

Queen Mary University of London, UK

Abstract

For several years, research at Queen Mary's Centre for Digital Music has probed the intersection of signal analysis technologies with informatics technologies. More specifically, we have built audio signal-level feature extractors which output RDF—the language of the Semantic Web—which have enabled us to build prototypes that expose enhanced functionality and offer new experiences to music users of all kinds. This talk will summarise some of our early and recent work in Semantic Audio and Music Informatics, leading up to the current research themes of our FAST-IMPACT (Fusing Audio and Semantic Technologies for Intelligent Music Production and Consumption) project. FAST-IMPACT is a 5 year programme of research involving 3 UK universities together with commercial and non-commercial partners from around the world. Its overarching aim is to bring more engaging and immersive experiences based on musical knowledge of all kinds to users of all kinds. Where relevant and possible, the principles will be illustrated with demos.

Biography

Mark Sandler was born in 1955. He received the B.Sc. and Ph.D. degrees from the University of Essex, U.K., in 1978 and 1984, respectively. His PhD was an investigation into Digital Audio Power Amplification and he has been an active researcher in Digital Audio and Digital Music ever since. He is a Professor of Signal Processing at Queen Mary University of London, London, U.K., where he founded the Centre for Digital Music and he has published over 400 papers in journals and conferences and supervised over 30 PhD students. Mark Sandler is currently Director of the EPSRC/AHRC Centre for Doctoral Training in Media and Arts Technology, and Principal Investigator of the 5 year research project, Fusing Audio and Semantic Technologies for Intelligent Music Production and Consumption. Mark is a Fellow of the Institute of Engineering and Technology (FIET), a Fellow of the Audio Engineering Society (FAES), a Fellow of the British Computer Society (FBCS), and a Fellow of the Institution of Electronic and Electrical Engineers (FIEEE).

Keynote Talk 2

The Promise of Music Information Retrieval: Are we there yet?

J. Stephen Downie

University of Illinois at Urbana-Champaign, USA

Abstract

In 1988, I needed to find some easy-to-play music for an upcoming flute performance exam. As a famously mediocre flute player, I was more than a little desperate to find something that would require the least amount of rehearsal time (and talent) to perform. I had previously played a baroque piece in the key of C that just might fit the bill. After too many hours of searching that included innumerable consultations with infinitely patient music librarians and fellow music students, we finally determined that the work in question was Bach's Flute Sonata in C, BWV 1033. This real-world use case inspired me to ask: "Why is it so difficult to identify, and then locate, a rather famous piece of music that I had actually played before?" In this talk, I reflect on just how far music information retrieval research (MIR) has come in the intervening 27 years in making access to music resources quick and simple. As I expound on my personal journey as an MIR researcher, I will note some of the twists in the road that I have surprised me. I will also offer up some commentary on lesser-travelled paths that we as community should explore.

Biography

J. Stephen Downie is the Associate Dean for Research and a Professor at the Graduate School of Library and Information Science at the University of Illinois at Urbana-Champaign. Downie is the Illinois Co-Director of the HathiTrust Research Center (HTRC). He is also Director of the International Music Information Retrieval Systems Evaluation Laboratory (IMIRSEL) and founder and ongoing director of the Music Information Retrieval Evaluation eXchange (MIREX). Stephen Downie was the Principal Investigator on the Networked Environment for Music Analysis (NEMA) project, funded by the Andrew W. Mellon Foundation. Furthermore, he is Co-PI on the Structural Analysis of Large Amounts of Music Information (SALAMI) project, jointly funded by the National Science Foundation (NSF), the Canadian Social Science and Humanities Research Council (SSHRC), and the UK's Joint Information Systems Committee (JISC). Stephen Downie has been very active in the establishment of the Music Information Retrieval (MIR) community through his ongoing work with the International Society for Music Information Retrieval (ISMIR) conferences. He was ISMIR's founding President and now serves on the ISMIR board. Professor Downie holds a BA (Music Theory and Composition) along with a Master's and a PhD in Library and Information Science, all earned at the University of Western Ontario, London, Canada.

Tutorials

Tutorial 1

Why Singing is Interesting

Simon Dixon (Queen Mary University of London, UK)

Masataka Goto (AIST, Japan)

Matthias Mauch (Queen Mary University of London, UK)

Abstract

This tutorial aims to introduce to the ISMIR community the exciting world of singing styles, the mechanisms of the singing voice, and provide a guide to representations, engineering tools, and methods for analyzing and leveraging it. The singing voice is arguably the most expressive of all musical instruments, and all popular music cultures around the world use singing. Across disciplines, a lot is known about singing culture and the intricate physiological and psychological mechanisms of singing, but this knowledge is not exploited enough in much of the music information retrieval literature. The three parts of the tutorial (one hour each) are designed to remedy this: an introduction to singing styles, techniques and forms around the world (including a short introduction to the psychology of singing), a practical guide to the analysis of singing using music informatics tools, and an overview over various systems for singing information processing. Our aim is for music information retrieval specialists to walk away with a newly sparked passion for singing, and ideas of how to use our knowledge of singing, and singing information processing, to create new, exciting research.

Tutorial 2

Addressing the Music Information Needs of Musicologists

Richard J. Lewis (Goldsmiths College, University of London, UK)

Ben Fields (Goldsmiths College, University of London, UK)

Tim Crawford (Goldsmiths College, University of London, UK)

Abstract

The music information needs of musicologists are not being met by the current generation of MIR tools and techniques. While evaluation has always been central to the practice of the music information retrieval community, the tasks tackled most often address the music information needs of recreational users, such as playlist recommendation systems; or are specified at a level which is not very relevant to the needs of music researchers, such as beat or key finding; or have focused on—and possibly even become over-fitted to—a narrow range of musical repertoire which doesn't cover musicological interests. In this tutorial we will present those music information needs through topics including at least the following: the metadata requirements of historical musicology; working with symbolic corpora; studying musical networks; passage-level audio search; and musical understandings of audio features. As well as these scheduled presentations and discussions, we will ask the attendees to submit suggestions of musicologically motivated research questions suitable for MIR during the course of the tutorial. These will then be reviewed and discussed during the conclusion of the tutorial. Finally, we have invited Meinard Müller to conclude the tutorial by outlining his view on the current state of MIR for musicology. We are aiming to enable attendees, as experts in their own areas of MIR, to find new applications of their tools and techniques that can also serve the needs of musicologists. Given the selection of MIR topics we intend to cover, this tutorial will be of particular interest to those working in: musical metadata; symbolic MIR; audio search; and graph analytics. We believe contemporary musicology to be a rich source of new and exciting challenges for MIR and we are confident the community can rise to those challenges. In the long term, we hope this tutorial will give rise to a selection of new MIREX tasks that focus on musicological challenges.

Tutorial 3

Markov Logic Networks for Music Analysis

Helene Papadopoulos (CNRS, Paris, France)

Abstract

The automatic extraction of relevant content information from music audio signals is an essential aspect of Music Information Retrieval (MIR). Music audio signals are very rich and complex, both because of the intrinsic physical nature of audio (incomplete and noisy observations, many modes of sound production, etc.), and because they convey multi-faceted and strongly interrelated semantic information (harmony, melody, metric, structure, etc.). Dealing with real audio recordings thus requires the ability to handle both uncertainty and complex relational structure at multiple levels of representation. Until recent years, these two aspects have been generally treated separately, probability being the standard way to represent uncertainty in knowledge, while logical representation being used to represent complex relational information. Markov Logic Networks (MLNs), in which statistical and relational knowledge are unified within a single representation formalism, have recently received considerable attention in many domains such as natural language processing, link-based Web search, or bioinformatics. The goal of this tutorial is to provide a comprehensive overview of Markov logic networks and show how they can be used as a highly flexible and expressive yet concise formalism for the analysis of music audio signals. We will show how MLNs encompass the probabilistic and logic-based models that are classically used in MIR. Algorithms for MLN modeling, training and inference will be presented, as well as open-source software packages for MLNs that are suitable to MIR applications. We will discuss concrete case-study examples in various fields of application.

Tutorial 4

COMputation and FLAmenco: Why Flamenco is Interesting for MIR Research

Emilia Gómez (Universitat Pompeu Fabra, Barcelona, Spain)

Nadine Kroher (Universitat Pompeu Fabra, Barcelona, Spain)

Jose Miguel Díaz-Báñez (Universidad de Sevilla, Spain)

Sergio Oramas (Universitat Pompeu Fabra, Barcelona, Spain)

Joaquín Mora (Universidad de Sevilla, Spain)

Francisco Gómez-Martín (Universidad Politécnica de Madrid, Spain)

Abstract

This tutorial provides an introduction to flamenco music with the support of MIR techniques. At the same time, the tutorial analyzes the challenges and opportunities that this music repertoire offers MIR researchers, presents some research contributions and provides a forum to discuss about how to address those challenges in future research. As ISMIR 2015 is in Málaga, this tutorial will give ISMIR participants a unique chance to discover flamenco music in its original location. The tutorial will be structured in two main parts. First, we will provide a general introduction to flamenco music: origins and evolution, musical characteristics, instrumentation, singing and guitar. We will illustrate this introduction with multimedia material and live performance. Then we will analyze how MIR technologies perform for flamenco music. By discussing several MIR tasks and how they should be addressed in this context, we will discover more about flamenco and how methods tailored to this repertoire can be exploited in other contexts. We will focus on automatic transcription, singer identification, music similarity, genre classification, rhythmic and melodic pattern detection and context-based music description methods. Participants will have the chance to interact with MIR annotated datasets and tools developed for flamenco music in the context of the COFLA project.

Tutorial 5

Using Correlation Analysis and Big Data to Identify and Predict Musical Behaviors

Jeff C. Smith (Smule)

Abstract

New and significant repositories of musical data afford unique opportunities to apply data analysis techniques to ascertain insights of musical engagement. These repositories include performance, listening, curation, and behavioral data. Often the data in these repositories also includes demographic and/or location information, allowing studies of musical behavior, for example, to be correlated with culture or geography. Historically, the analysis of musical behaviors was limited. Often, subjects (e.g. performers or listeners) were recruited for such studies. This technique suffered from issues around methodology (e.g. the sample set of subjects would often exhibit bias) or an insufficient number of subjects and/or data to make reliable statements of significance. That is to say the conclusions from these studies were largely anecdotal. In contrast to these historical studies, the availability of new repositories of musical data allow for studies in musical engagement to develop conclusions that pass standards of significance, thereby yielding actual insights into musical behaviors. This tutorial will demonstrate several techniques and examples where correlation and statistical analysis is applied to large repositories of musical data to document various facets of musical engagement. Web site: <https://ccrma.stanford.edu/damp/> Stanford University has created a new corpus of amateur music performance data, the Stanford Digital Archive of Mobile Performances, or DAMP, to facilitate the study of musical engagement through application of correlation and statistical analysis.

Tutorial 6

Automatic Music Transcription

Zhiyao Duan (University of Rochester, USA)

Emmanouil Benetos (Queen Mary University of London, UK)

Abstract

Automatic Music Transcription (AMT) is a fundamental problem in music information retrieval. Roughly speaking, transcription refers to extracting a symbolic representation—a list of notes (pitches and rhythms)—from an audio signal. Music transcription is a fascinating but challenging task, even for humans: in undergraduate music education it is usually called dictation, and achieving a high level of proficiency requires years of practice and training. Empowering machines with this ability is an even more challenging problem, especially for automatically transcribing polyphonic music. To that end, the AMT problem has drawn great interest of researchers from several areas including signal processing, machine learning, acoustics, music theory, and music cognition. In terms of applications, a successful AMT system would be helpful for solving many MIR research problems, including music source separation, structure analysis, content-based music retrieval, and musicological study of non-notated music, just to name a few. This tutorial will give an overview of the AMT problem, including current approaches, datasets and evaluation methodologies. It will also explore connections with other related problems (i.e. audio-score alignment, source separation) as well as applications to related fields, such as content-based music retrieval and computational musicology. The tutorial is designed for students and researchers who have general knowledge of music information retrieval and/or computational musicology and are interested in getting into the field of AMT. A substantial amount of time will be spent in discussing challenges and research directions; we hope that this discussion will help move this field forward, and influence related fields in MIR and computational musicology to exploit AMT technologies. The tutorial will also include hands-on sessions on using AMT code and plugins - participants will be encouraged to bring their laptops and gain access to transcription datasets, as well as work on AMT examples.

Poster Session 1

IMAGE QUALITY ESTIMATION FOR MULTI-SCORE OMR

Dan Ringwalt, Roger B. Dannenberg

Carnegie Mellon University

School of Computer Science

ringwalt@cmu.edu, rbd@cs.cmu.edu

ABSTRACT

Optical music recognition (OMR) is the recognition of images of musical scores. Recent research has suggested aligning the results of OMR from multiple scores of the same work (multi-score OMR, MS-OMR) to improve accuracy. As a simpler alternative, we have developed features which predict the quality of a given score, allowing us to select the highest-quality score to use for OMR. Furthermore, quality may be used to weight each score in an alignment, which should improve existing systems' robustness. Using commercial OMR software on a test set of MIDI recordings and multiple corresponding scores, our predicted OMR accuracy is weakly but significantly correlated with the true accuracy. Improved features should be able to produce highly consistent results.

1. INTRODUCTION

Optical music recognition (OMR) is the problem of converting scanned music scores into a symbolic format such as MIDI. The advantages of OMR for computer music applications are clear, but it has yet to be widely used in many applications which use MIDI or MusicXML scores. Although OMR has been studied extensively since the 1960s, no OMR system has near-perfect accuracy. Commonly, the output of an OMR system must be checked by hand and at least a few corrections must be made, making the process extremely time-consuming [2]. This limits the amount of music which may be digitized, and in fact, much music is still digitized completely by hand in sources such as the Mutopia Project [1]. Recent research has focused on using contextual information beyond what is present on a single page to improve OMR results.

Recently, the Petrucci Music Library (or International Music Score Library Project, IMSLP) [17] has become a high-quality source of public domain music scores. The site allows users to scan and upload scores. Therefore, there may be several scores of the same work, which may be musically identical, or different editions, arrangements, or parts. There is a large discrepancy between the scanning equipment each user has, along with their relative care

in scanning, so image quality varies widely. At the time of writing, IMSLP contains over 90,000 works, for which there are over 300,000 uploaded scores.

Although many OMR errors are due to notational complexity [2], we expect at least some mistakes to be due to random deformation in the score, independent of the content. Then if multiple scores are available corresponding to one piece, a consensus built from OMR applied to each score should be more accurate than any one score. The possibility of aligning multiple scores of the same work to build a single result (multi-score OMR, MS-OMR) is already being explored [21].

However, scores available from IMSLP and other sources vary widely in noise introduced in the scanning process. Previous work on multi-recognizer OMR (MR-OMR), where the results are aligned from several OMR systems on the same score, has noted that a consensus result using simple voting may be worse than the result of the best recognizer [4]. Similarly, if there are several poor scores for a work and one good score, a MS-OMR result may be worse than the result on the highest-quality score alone. An MS-OMR system that correctly estimates the quality of each score and acts accordingly should overcome this limitation.

Formally, we want to predict some accuracy measure of OMR, given features extracted from an image. We define the *quality* of an image to be the predicted accuracy given by our resulting model. Quality should depend on factors such as random noise, deformation of the page, and resolution, and is expected to be correlated with OMR accuracy. Our predicted value is mostly useful in comparisons between scores; even if the actual accuracy is on a 0 to 1 scale, a quality value learned using linear regression may be outside this range for some scores, and so it may not be interpretable as an accuracy value. However, even if we evaluate multiple recognizers using the same methodology, then we can learn a separate quality value for each recognizer, and predict the best-performing recognizer for a new score.

Clearly, the quality value gives useful information to a MS-OMR system. We may want to throw out some scores altogether if their quality is too low, as they may not contribute much of a benefit in addition to the higher-quality scores. As a simplification, we may only take the highest-quality score, and perform normal OMR. If our quality value is accurate, then this is the safest approach, because by introducing other scores, we risk lowering the accuracy.



© Dan Ringwalt, Roger B. Dannenberg.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Dan Ringwalt, Roger B. Dannenberg. "Image Quality Estimation for Multi-Score OMR", 16th International Society for Music Information Retrieval Conference, 2015.

This is clearly less computationally expensive than obtaining and aligning multiple OMR results, but should result in much higher accuracy than randomly choosing any available score. We consider this approach to MS-OMR in this paper.

2. RELATED WORK

2.1 Multi-Recognizer and Multi-Image OMR

Recent research has focused on improving OMR accuracy by aligning several OMR results and building a consensus score. Byrd and Schindele [5] designed a multi-recognizer OMR system which applies multiple OMR systems to the same score, and resolves conflicts between results using pre-defined rules. This is built on the assumption that each OMR system will have particular situations in which it outperforms the other systems. As OMR systems are under development and their strengths and weaknesses may change, a system is proposed which automatically learns the performance of each system in different possible situations.

More recently, Bugge et al. [4] proposed another multi-recognizer OMR system that resolves conflicts between each recognizer by a simple majority vote. Scores are exported as MusicXML from each recognizer, and converted to a custom subset of MusicXML, "MusicXiMpLe," which only stores the information necessary to decode note pitch and duration.

Padilla et al. have suggested extending multiple-recognizer OMR to align the results from multiple images of the same score [21]. A method is proposed to profile the response of each OMR tool to score quality, by adding additional noise to existing scores with available ground truth and measuring OMR accuracy.

2.2 Image Quality

Existing measures have been designed to estimate the level of degradation present in an image due to the scanning process. Kanungo et al. developed a local distortion model (referred to as *Kanungo noise*) for binary images which is an extension of simple salt-and-pepper noise, and uses 6 parameters [15]. The additional parameters capture the increased noise near the boundary between black and white pixels, and correlation in noise between nearby pixels.

Kanungo et al. previously estimated the Kanungo noise parameters of a binary image of a text document [16]. The estimation requires an ideal set of synthetic text documents with similar font face and size to the scanned image. Given an estimated set of parameters, each ideal image is degraded using the parameters. All 3x3 square patterns are found in each degraded ideal image and the input image, and a histogram for the count of each of $2^{3*3} = 512$ patterns is made for the degraded ideal images and the input. A Kolmogorov-Smirnov test statistic is measured between the cumulative distribution functions of both histograms. This statistic is minimized using the Nelder-Mead simplex method [19].

Additionally, prior work in OMR has focused on undoing global distortions present in the input image. The level of distortion detected by these methods is another feature which should be negatively correlated with OMR accuracy. For example, Fujinaga's staff detection algorithm [12] tries to correct bending of the staves due to page curl. This *deskewing* process translates each column of the image to make the staff lines more horizontal. We use the mean vertical translation performed by deskewing as one feature.

We may also robustly estimate the resolution of an image using the distance between staff lines. Unlike the actual size of the image, this does not depend on the size of the original page, and all symbols such as notes will be directly proportional to the staffline distance. We use Cardoso et al.'s robust estimated staffline distance [7] as another feature.

3. METHODS

3.1 Data Acquisition

All available scores of Ludwig van Beethoven's piano sonatas were obtained from IMSLP. In total, there were 32 sonatas, with 285 different scores.

MIDI versions of several movements from the Beethoven piano sonatas were obtained from the Mutopia Project [1], and served as ground truth to compare with the OMR results. The MIDI version was automatically generated from a manually transcribed LilyPond [20] source file.

As the MIDI files are separated by movement, the scores were also split into each movement. Therefore, each *work* is defined to be a single movement of a sonata.

3.2 Score Preprocessing

The scores were preprocessed by a custom system before extracting image quality features and performing OMR. Our methods for rotation correction and staff and staff system detection are described in [25].

Many scores had movements which started in the middle of the page. Therefore, the staff systems which formed the start of each movement were labeled by hand. Our system was used to automatically segment pages as necessary to split the score into movements.

We kept 67 original scores from IMSLP which contained an entire sonata and were not an arrangement or other version, and had ground truth for at least one movement available from the Mutopia Project. We successfully generated and processed 95 single-movement scores for 16 works (single movements), belonging to 8 different sonatas.

3.3 Image Quality Features

Kanungo parameter estimation was performed on each pre-processed page. A page from a LilyPond-engraved score obtained from the Mutopia Project was used as the ideal image. Each image was scaled to a normalized staffline

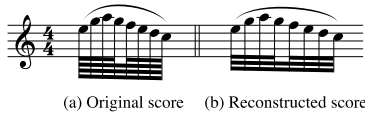


Figure 1. This error only counts as one error under low-level evaluation, but several under high-level evaluation, as the length of each note is incorrect. Source: [24]

distance value of 8. Kanungo noise parameters were estimated using the SciPy [13] implementation of Nelder-Mead optimization [19], as described in Section 2.2. Nelder-Mead was run 10 times starting from a uniformly random parameter distribution, and was stopped after 50 function evaluations each time. The resulting Kanungo parameters ($\nu, \alpha_0, \alpha, \beta_0, \beta, k$) were used as features to predict OMR performance.

We also performed Fujinaga’s staff detection algorithm, which skews the image to correct page curl. This gives us the amount of page curl in the original image. We use the mean vertical translation performed by this deskewing as one feature, which represents the degree of distortion in the page.

Finally, we used Cardoso et al.’s robust staffline distance estimation method [7]. We used the staffline distance, and the ratio of staffline thickness to distance, as two more features. The staffline distance represents the resolution of the image, while the thickness-to-distance ratio represents the relative thickness of lines on the page.

3.4 OMR

The preprocessed movements were processed by the SharpEye 2 OMR system, version 2.68. The result was exported to MIDI.

4. EVALUATION

4.1 OMR Evaluation Methods

OMR researchers have yet to adopt any evaluation metric as a common standard [6], and specialized evaluation methods will likely be needed for most systems. We chose as basic of an evaluation method as possible: simply comparing the start time of each note to the ground truth. This still requires rests, accidentals, and other basic symbols to be detected correctly in the usual case; it cannot detect a too-short note followed by a too-long rest, but this particular error should be extremely rare. Although it does not test other information like dynamic markings, we consider these to be of secondary importance compared to the actual notes. As we only consider the start position of each note, and not the duration of notes and rests, our evaluation is a further simplification of previous evaluations, which consider both the start and end of notes [4, 14].

SharpEye 2 outputs a proprietary .mro format which contains information such as the position of some individ-

ual symbols. Therefore, it is possible to conduct a *low-level* evaluation if the score is labeled with the position of each symbol. Although both values should be highly correlated, high-level accuracy may decrease drastically with only a small decrease in low-level accuracy, as illustrated in Figure 1.

Our evaluation method is considered high-level. This allows us to use MIDI recordings from the Mutopia Project, which only contain the actual notes, as our labeled data. One potential issue with MIDI is that to simulate a realistic performance, staccato notes may have a shortened length followed by a rest for their remaining time. Our evaluation, which only tests the start of each note, accounts for this.

4.2 Accuracy Value

Given two aligned scores, we need to derive a single value for the accuracy. Here, each note is represented as the time in the score, and a pitch, and a note is correctly detected if there is a note with the exact same values in the original score. The OMR output may contain both false positives, where a note is accidentally detected, and false negatives, where a note is missing. We may calculate the precision p , which is the proportion of true positives to all detected notes, and the recall r , which is the proportion of true positives to all notes in the original score. The standard method of combining these values, which we use as our accuracy value, is the F_1 score:

$$F_1 = \frac{2pr}{p+r}$$

4.3 MIDI-MIDI Alignment

All MIDI files were imported into Python using music21[8]. Next, we aligned each OMR output to the ground truth, to correct for missing or extra measures due to OMR errors. We noticed that LilyPond’s MIDI output (used by Mutopia) pads a pickup measure to the length of a full measure, while SharpEye 2’s does not. Therefore, we align each beat rather than each measure, so that the pickup will also be correctly aligned.

The standard alignment algorithm, used in both bioinformatics and computer music applications, is Needleman-Wunsch [18, 3]. It minimizes the sum of the distance between each aligned element of two sequences, plus a penalty for each inserted gap. In our case, our distance matrix has one row for each beat in the real score, and one column for each beat in the OMR score. The distance entry for each pair is $1 - F_1$ for the pair of beats, multiplied by the maximum of the number of notes in both beats. (This is implicitly 0 when both beats only contain rests, and the F_1 score would normally be undefined.) We use a gap penalty of 10.

After Needleman-Wunsch, we simply calculate the F_1 score for the entire aligned scores, with new positions for the notes accounting for inserted gaps. This is our OMR accuracy value.

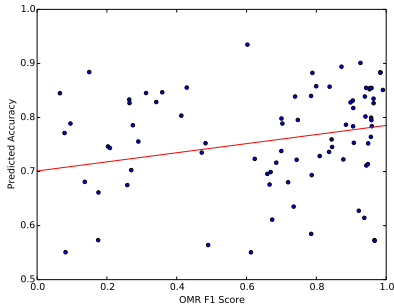


Figure 2. The OMR F_1 score for each movement compared with the predicted accuracy, with the best-fit line.

4.4 Quality Estimation

We used a linear model to predict the OMR F_1 score given our features. We chose the Scikit-learn [22] implementation of Support Vector Regression with a linear kernel, as it seemed to perform better than ordinary least squares linear regression. The model was validated by leave-one-out testing on each work: for each work, a model was trained excluding its corresponding scores, and the predicted best score for the work was compared to the score with the highest accuracy. Finally, we fit the model to the entire dataset to determine the coefficients.

5. RESULTS

The OMR F_1 score and predicted accuracy were weakly but significantly correlated ($R = 0.30, p = 0.0029$). The data is shown in Figure 2.

For each work, we compared the score with the highest OMR accuracy and the score with the highest predicted quality using leave-one-out testing (Table 1). Six of the 16 works had a correctly predicted best score, whereas using uniformly random guessing, the expected number of correct scores is only 2.82. The full OMR accuracy results are presented in Table 2.

We also noted that the best few scores may all have nearly the same high accuracy. In these cases, it is not necessary that our top predicted score has the highest accuracy, but the accuracy should be close to the highest. For each work, we considered the mean accuracy of all scores, which is the expected accuracy of a score selected by random choice, the highest accuracy, and the accuracy of the predicted best score. The mean of the expected accuracy for random guessing is 0.61, and the mean of the best accuracy (the best possible result) is 0.82, while the mean accuracy of the best predicted scores is 0.74. The chosen score’s accuracy was higher than expected in 14 of 16 cases. This confirms that our method reliably outperforms random guessing, but there is still room to improve

Work	Best	Pred. Best	# Scores
1.1	IMSLP66390	IMSLP05524	8
1.4	IMSLP66390	IMSLP05524	7
5.1	IMSLP66394	IMSLP66394	5
5.3	IMSLP66394	IMSLP66394	5
6.3	IMSLP66395	IMSLP66395	5
19.1	IMSLP00019	IMSLP04073	6
19.2	IMSLP05545	IMSLP69581	6
20.1	IMSLP45469	IMSLP66410	7
20.2	IMSLP05546	IMSLP05546	7
23.2	IMSLP51795	IMSLP04078	3
23.3	IMSLP66412	IMSLP66412	6
25.1	IMSLP66414	IMSLP66414	6
25.2	IMSLP66414	IMSLP69588	6
25.3	IMSLP66414	IMSLP69588	6
27.1	IMSLP66416	IMSLP05553	6
27.2	IMSLP69590	IMSLP05553	6

Table 1. Accuracy predictions on the Beethoven piano sonata test set. For each work (identified by *sonata number.movement*), we compare the score with the highest OMR accuracy (*Best*) and the highest predicted quality (*Pred. Best*).

in choosing one of the best scores.

The coefficients of our linear model (Table 3 in the appendix) are directly interpretable as the effect each parameter has on OMR accuracy. Many results were unexpected. For example, ν represents the probability of salt-and-pepper noise in the Kanungo model, which should negatively affect OMR accuracy, but its coefficient is positive. However, as it is on a small scale (typically 0 – 0.05), it has a smaller impact on accuracy. This result may be due to a few outliers which had poor results for Kanungo estimation.

The coefficient for `mean_skew`, which is the deformation undone by Fujinaga’s deskewing, is also unexpectedly positive. This may indicate a flaw in our implementation, or again, outliers. We did find that `staff_dist` is positively correlated with accuracy, as we expect that higher-resolution scores will have better results. The coefficient is small, but more significant as `staff_dist` is on a larger scale (usually at least 20).

6. CONCLUSIONS

We introduced an estimated OMR accuracy measure, and showed that its correlation to the true accuracy is statistically significant. However, the correlation is too low to correctly predict the best-quality score a majority of the time. On the other hand, this validates the use of features extracted from the image to select higher-quality scores. By refining our features and adding additional ones, we should be able to build a practical quality estimation system which can support multi-score OMR.

Since most of our current image features are parameters for Kanungo noise, the success of the image quality estimation is dependent on these parameters being accu-

rate. The Kanungo estimation process is currently very time-intensive, requiring around 2 minutes per page. Some instances of Nelder-Mead will become stuck in a local optimum, so repeating Nelder-Mead even more times should improve results. However, the time involved made this impractical in this case.

7. FUTURE DIRECTIONS

7.1 Image Quality Features

We only found a weak correlation between OMR accuracy and predicted quality, and noted that the Kanungo parameter estimates were noisy. Furthermore, the estimation process is too slow to be practical for a large music library such as IMSLP. Therefore, a better performing, faster Kanungo estimation process is needed to make image quality estimation practical.

We may be able to improve Kanungo estimation by using assumptions specific to music scores, which would allow us to test a much smaller area of the image. For example, if we find all empty stretches of staff on the page, we can concatenate some of these as the input to Kanungo estimation. We may generate an ideal empty staff using the estimated staffline distance and thickness. This uses a much smaller image, and may even be more robust as differences in typography between the ideal and input image will not affect it.

Finally, our features only take into account errors introduced in the scanning process. However, differences in the original score, such as different fonts, should also affect the accuracy of a particular OMR recognizer. *Adaptive* OMR systems [11, 23] improve their performance on scores with a certain font and other particularities by learning from their corrected output. If an adaptive system is trained using a homogenous set of scores with a particular font, then we may be able to extract information about the font from its classification model. Features which have been used for handwritten music writer identification [10] may be useful.

7.2 OMR Evaluation

We mentioned that a small difference in low-level accuracy may make a dramatic difference in high-level accuracy. Therefore, low-level accuracy may be a more stable value to use when performing regression. However, obtaining a real-world test set of a similar size with low-level ground truth would be much more time-consuming.

Using scores from the Mutopia Project, it would be possible to modify LilyPond to output the position of each symbol, giving us a low-level ground truth. Next, we could apply deformations such as Kanungo noise to the output before performing OMR. This is similar to Padilla et al.'s proposal to add additional noise to real images from IMSLP to profile each OMR recognizer. However, if we start from ideal computer-engraved images, then the parameters we use to add noise to the image are exactly the same as our image quality features. Therefore, we may design our test set to cover the entire parameter space, and we can

directly learn our image quality function using regression from the input parameters to the OMR accuracy for each recognizer.

On the other hand, we may be able to improve our results while keeping high-level accuracy. We may obtain a broader range of scores from IMSLP paired with MIDI recordings from the Mutopia Project, which would provide us with more training data. Using more data, we could train a more sophisticated model than linear regression, which would hopefully better predict accuracy. We noted that a single error has a proportional effect in low-level accuracy but a much bigger effect on high-level accuracy, so high-level accuracy likely has a nonlinear relationship with quality. Therefore, methods such as kernel SVR or random forests may be able to capture this nonlinear relation.

We noticed that some MIDI scores were unable to be opened by music21, and they were excluded from the analysis. This is believed to be because some note durations cannot be unambiguously converted from a floating-point time value back to the music-theoretic note values which music21 uses. This should be possible to fix by using the MIDI files in their original form, which would allow us to include more data in our analysis.

7.3 Alignment-Based MS-OMR

Although we presented our method as a simpler alternative to existing MS-OMR systems, our image quality estimate may be used in a larger system. An MS-OMR system which aligns multiple results, as in [21], may be augmented by weighting each score by its quality in the vote. Furthermore, alignment-based MS-OMR systems require a multiple sequence alignment, and finding the globally optimal such alignment is NP-complete [26]. Approximate multiple alignment algorithms often use a series of pairwise alignments [9]. Recent research in aligning multiple musical recordings or scores used a progressive alignment, where pairwise alignments were performed sequentially on the inputs [27, 4]. Ordering OMR results from highest to lowest quality may work better than other orders.

We have demonstrated the usefulness of image quality estimation in predicting OMR accuracy. A more robust quality estimate should be useful for any MS-OMR system. This should have a significant impact on OMR accuracy for large music libraries such as IMSLP.

Work	Score	Accu.	Qual.	Score	Accu.	Qual.	Score	Accu.	Qual.	Score	Accu.	Qual.	Score	Accu.	Qual.
1.1	00001	0.95	0.62	03796	0.70	0.79	05524	0.95	0.88	51707	0.57	Error	66279	0.70	0.75
1.1	66390	0.96	0.79	77993	0.61	0.50	90564	0.08	0.18	243106	0.84	0.83			
1.4	00001	0.94	0.56	03796	0.79	0.66	05524	0.31	0.82	51707	0.37	Error	66279	0.78	0.85
1.4	66390	0.96	0.86	77993	0.62	0.70	243106	0.67	0.65						
5.1	00005	0.92	0.82	02412	0.08	Error	03858	0.81	0.69	51714	0.85	Error	66394	0.96	0.83
5.1	68715	0.69	0.69	243114	0.91	0.75									
5.3	00005	0.17	0.67	03858	0.49	0.46	51714	0.18	Error	66394	0.60	0.87	68715	0.47	0.71
5.3	243114	0.17	0.53												
6.3	00006	0.41	0.79	03859	0.34	0.80	51715	0.40	Error	66395	0.43	0.85	68719	0.36	0.68
6.3	243121	0.10	0.80												
19.1	00019	0.75	0.72	04073	0.15	0.89	05545	0.26	0.76	45370	0.26	0.62	51743	0.20	0.73
19.1	66408	0.28	Error	69581	0.72	0.62	345618	0.27	Error						
19.2	00019	0.91	0.75	04073	0.84	0.74	05545	0.94	0.84	45370	0.94	0.78	51743	0.67	0.57
19.2	66408	0.98	Error	69581	0.93	0.93	345618	0.94	Error						
20.1	00020	0.08	0.66	04075	0.85	0.75	05546	0.96	0.83	45469	0.97	0.52	51745	0.67	0.54
20.1	66410	0.07	0.82	69582	0.90	0.85									
20.2	00020	0.95	0.59	04075	0.14	0.64	05546	0.98	0.88	45469	0.95	0.74	51745	0.79	0.50
20.2	66410	0.08	0.50	69582	0.94	0.70									
23.2	03184	0.11	0.51	04078	0.46	0.70	51795	0.55	0.59						
23.3	00023	0.57	0.67	03184	0.09	0.55	04078	0.38	0.72	05549	0.58	0.78	51795	0.41	0.53
23.3	66412	0.60	0.86												
25.1	00025	0.97	0.52	03185	0.43	Error	04081	0.80	0.83	05551	0.96	0.76	51797	0.88	0.68
25.1	66414	0.98	0.88	69588	0.26	0.84									
25.2	00025	0.84	0.73	04081	0.73	0.57	05551	0.95	0.85	51797	0.74	0.64	66414	0.99	0.68
25.2	69588	0.87	0.89												
25.3	00025	0.92	0.56	04081	0.66	0.63	05551	0.96	0.79	51797	0.74	0.70	66414	0.96	0.74
25.3	69588	0.94	0.84												
27.1	00027	0.88	0.75	04090	0.79	0.86	05553	0.90	0.83	51799	0.48	0.76	66416	0.91	0.81
27.1	69590	0.70	0.78												
27.2	00027	0.27	0.69	04090	0.21	0.70	05553	0.27	0.75	51799	0.18	0.60	66416	0.29	0.74
27.2	69590	0.51	0.55												

Table 2. OMR accuracy (F_1) values for each score (by IMSLP ID), and predicted quality values.

Variable	Coefficient	Variable	Coefficient
ν	4.2	mean_skew	19.34
α_0	1.7	staff_dist	0.021
α	0.10	staff_thick_ratio	0.22
β_0	-0.70		
β	-0.077		
k	-0.0026		

Table 3. Coefficients of the linear model for image quality.

8. REFERENCES

- [1] The Mutopia Project, 2015. <http://mutopiaproject.org/> (accessed July, 2015).
- [2] D. Bainbridge and T. Bell. The challenge of optical music recognition. *Computers and the Humanities*, 35(2):95-121, 2001.
- [3] J. J. Bloch and R. B. Dannenberg. Real-time accompaniment of polyphonic keyboard performance. In *Proceedings of the 1985 International Computer Music Conference*, pages 279-290, 1985.
- [4] E. P. Bugge, K. L. Juncher, B. S. Mathiasen Jakob, and J. G. Simonsen. Using sequence alignment and voting to improve optical music recognition from multiple recognizers. In *Proceedings of the 12th International Conference on Music Information Retrieval*, pages 405-410, 2001.
- [5] D. Byrd and M. Schindele. Prospects for improving OMR with multiple recognizers. In *Proceedings of the 7th International Conference on Music Information Retrieval*, pages 41-46, 2006.
- [6] D. Byrd and J. G. Simonsen. Towards a standard testbed for optical music recognition: definitions, metrics, and page images, 2015. <http://www.informatics.indiana.edu/donbyrd/OMRTestbed/OMRStandardTestbed1Mar2013.pdf> (accessed July, 2015).
- [7] J. S. Cardoso and A. Rebelo. Robust staffline thickness and distance estimation in binary and graylevel music scores. In *20th International Conference on Pattern Recognition*, pages 1856-1859, 2010.
- [8] M. S. Cuthbert and C. Ariza. music21: A toolkit for computer-aided musicology and symbolic music data. In *Proceedings of the 11th International Conference on Music Information Retrieval*, pages 637-42, 2010.
- [9] R. C. Edgar. MUSCLE: Multiple sequence alignment with high accuracy and high throughput. *Nucleic acids research*, 32(5):1792-7, 2004.
- [10] A. Fornés, A. Dutta, A. Gordo, and J. Lladós. The ICDAR 2011 music scores competition: staff removal and writer identification. In *2001 International Conference on Document Analysis and Recognition*, pages 1511-15, 2011.
- [11] I. Fujinaga. *Adaptive Optical Music Recognition*. PhD thesis, McGill University, 1996.
- [12] I. Fujinaga. Staff detection and removal. In *Visual Perception of Music Notation: On-Line and Off-Line Recognition*, pages 1-39, 2004.
- [13] E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, 2001. <http://www.scipy.org/> (accessed July, 2015).
- [14] G. Jones, B. Ong, I. Bruno, and K. Ng. Optical music imaging: music document digitisation, recognition, evaluation, and restoration. In *Interactive Multimedia Music Technologies*, pages 50-79, 2008.
- [15] T. Kanungo, R. M. Haralick, and I. Phillips. Global and local document degradation models. In *Proceedings of the Second International Conference on Document Analysis and Recognition*, pages 730-734, 1993.
- [16] T. Kanungo and Q. Zheng. Estimation of morphological degradation model parameters. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings*, volume 3, pages 1961-1964, 2001.
- [17] Project Petrucci LLC. IMSLP/Petrucci Music Library, 2015. <http://imslp.org/> (accessed July, 2015).
- [18] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of molecular biology*, 48(3):443-453, 1970.
- [19] J. A. Nelder and R. Mead. A simplex method for function minimization. *The computer journal*, 7(4):308-313, 1965.
- [20] H.-W. Nienhuys and J. Nieuwenhuizen. LilyPond, a system for automated music engraving. In *Proceedings of the XIV Colloquium on Musical Informatics (XIV CIM 2003)*, pages 1-6, 2003.
- [21] V. Padilla, A. Marsden, A. McLean, and K. Ng. Improving OMR for digital music libraries with multiple recognizers and multiple sources. In *Proceedings of the 1st International Workshop on Digital Libraries for Musicology - DLfM '14*, pages 1-8, 2014.
- [22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Courapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825-2830, 2011.
- [23] L. Pugin, J. A. Burgoyne, and C. Ha. MAP adaptation to improve optical music recognition of early music documents using hidden Markov models. In *Proceedings of the 8th International Conference on Music Information Retrieval*, pages 513-16, 2007.
- [24] T. Reed. *Optical music recognition*. Master's thesis, University of Calgary, 1995.
- [25] D. Ringwalt, R. B. Dannenberg, and A. Russell. Optical music recognition for live score display. In *Proceedings of the 2015 Conference on New Interfaces for Musical Expression*, 2015.
- [26] L. Wang and T. Jiang. On the complexity of multiple sequence alignment. *Journal of Computational Biology*, 1(4):337-348, 1994.
- [27] S. Wang and S. Dixon. Robust joint alignment of multiple versions of a piece of music. In *Proceedings of the 15th International Conference on Music Information Retrieval*, pages 83-88, 2014.

COMPARATIVE MUSIC SIMILARITY MODELLING USING TRANSFER LEARNING ACROSS USER GROUPS

Daniel Wolff, Andrew MacFarlane and Tillman Weyde

Music Informatics Research Group – Department of Computer Science
City University London

{daniel.wolff.1, a.macfarlane-1, t.e.weyde}@city.ac.uk

ABSTRACT

We introduce a new application of transfer learning for training and comparing music similarity models based on relative user data: The proposed Relative Information-Theoretic Metric Learning (RITML) algorithm adapts a Mahalanobis distance using an iterative application of the ITML algorithm, thereby extending it to relative similarity data. RITML supports transfer learning by training models with respect to a given template model that can provide prior information for regularisation. With this feature we use information from larger datasets to build better models for more specific datasets, such as user groups from different cultures or of different age. We then evaluate what model parameters, in this case acoustic features, are relevant for the specific models when compared to the general user data.

We to this end introduce the new CASimIR dataset, the first openly available relative similarity dataset with user attributes. With two age-related subsets, we show that transfer learning with RITML leads to better age-specific models. RITML here improves learning on small datasets. Using the larger MagnaTagATune dataset, we show that RITML performs as well as state-of-the-art algorithms in terms of general similarity estimation.

1. INTRODUCTION

Music similarity models are a central part of many applications in music research, particularly Music Information Retrieval (MIR). When training similarity models, it turns out that learnt models vary considerably for different data sets and application scenarios. Recently, context-sensitive models have been introduced, e.g. for the task of music recommendation (Stober [9] provides an overview). The main problem with context-sensitive similarity models is currently to obtain enough data to train the models for each context. Transfer learning promises to enable effective training of models for specific contexts by including information from related datasets. We here present an

approach of transfer learning in music similarity that improves results of specialised models, using our W_0 -RITML extension of Information-Theoretic Metric Learning (ITML). The template-based optimisation in W_0 -RITML allows for a comparison of the general and specialised models – it derives the latter from the former – which we suggest as a tool for comparative analysis of similarity data by (e.g. cultural) provenance.

We are particularly interested in modelling relative similarity ratings collected from participants during Games With a Purpose (GWAPs). Using similarity data from user groups promises to provide tailored model performance and the opportunity to compare such groups via the trained similarity models. The new CASimIR dataset presented in Section 3 contains such similarity ratings and information about the contributing subjects. We use this extra data to group users and here exemplarily train age-specific music similarity models based on age-bounded subsets. However, the relatively small size of the CASimIR dataset requires a different approach to training the group-specific models as existing algorithms are not sufficiently effective for this purpose.

We contribute a solution to this problem with a novel generic algorithm for transfer learning with similarity models: The RITML algorithm (see Section 5.2) extends on ITML to allow for learning a Mahalanobis metric from relative similarity data like in CASimIR. With W_0 -RITML, information learnt from remaining data can be successfully transferred to an age-bounded dataset via a Mahalanobis matrix. This transfer-learning increases performance on small datasets and provides interpretable values in the Mahalanobis matrix. The Mahalanobis matrix provides a compact representation of similarity information in a dataset. This is useful in scenarios where the music data is difficult to access due to its data volume or copyright restrictions. The CASimIR dataset and code used in this paper are available online¹.

2. RESEARCH BACKGROUND

Transfer learning relates to many areas and approaches in machine learning. A general overview of transfer learning is given in Pan and Yang [6]. In their categorisation, our task is an inductive knowledge transfer from one similarity modelling task to another via model parameters. Note that



© Daniel Wolff, Andrew MacFarlane and Tillman Weyde. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Daniel Wolff, Andrew MacFarlane and Tillman Weyde. "Comparative Music Similarity Modelling using Transfer Learning Across User Groups", 16th International Society for Music Information Retrieval Conference, 2015.

¹ <http://mirg.city.ac.uk/datasets/ismir2015dw>

in our example the tasks differ only in the dataset, but our method can also be used for more divergent tasks.

In MIR, transfer learning is a relatively new method. In 2013, [2] described multi-task learning using a shared latent representation for auto-tagging, genre classification and genre-based music similarity. This representation includes both the features and the labels for the different tasks. In experiments on several datasets they showed improvement of classification accuracy and modelling similarity according to genre.

We here work with relative similarity ratings from humans in our new CASimIR dataset for group-specific modelling. Furthermore, we use the MagnaTagATune dataset [3] for comparison on non-specific similarity learning. Here, the Support Vector Machine (SVM) approach developed by Schultz and Joachims [7] and applied in [10, 11] is used as state-of-the-art baseline.

Another state-of-the-art algorithm for learning from relative similarity data is Metric Learning To Rank (MLR). McFee et al. [4] introduce MLR for parametrising a linear combination of content-based features using collaborative filtering data. Their post-training analysis of feature weights revealed that tags relating to genre or radio stations were assigned greater weights than those related to music theoretical terms.

3. A DATASET FOR USER-AWARE SIMILARITY

In order to perform a related analysis and comparisons of models between different user groups, we have collected the CASimIR datasets using Spot the Odd Song Out [13], an online² multi-player Game With a Purpose (GWAP). The similarity module of the Spot the Odd Song Out game collects relative similarity data using an odd-one-out survey: From a set of three music clips, participants are asked to choose the clip most dissimilar to the remaining clips, i.e. the *odd song out*. The game motivates players by rewarding blind agreement. For various reasons, including personal data protection, little music annotation data is publicly available with information about the provider of the data and their context.

Although the game can collect anonymised personal information including gender, nationality, spoken languages and musical experience, the amount and type information available varies between participants, as data provision is voluntary. Our overarching goal is to study the relation between similarity and culture and we thus link annotations to cultural profiles rather than indexing specific participants. With this paper we publish the first set of similarity data with anonymised profiles.

3.1 Constraints from Relative Similarity Ratings

The MagnaTagATune and CASimIR datasets both contain relative similarity ratings. A participant's rating of C_k as

² <http://mirg.city.ac.uk/camir/game/>

the odd one out (of the triplet C_i, C_j, C_k) results in 2 relative similarity constraints: clips C_i and C_j are more similar than C_i and C_k , and clips C_j and C_i are more similar than C_j and C_k . These constraints are denoted as (i, j, k) and (j, i, k) , respectively which are contained in the constraint set \hat{Q} .

Human ratings regularly produce inconsistent constraints. We use the graph representation of the similarity data as suggested by [5] to analyse and filter inconsistencies: Each constraint (i, j, k) is represented by an edge connecting two vertices (i, j) $\xrightarrow{\alpha_{ijk}}$ (i, k) corresponding to two clip pairs, with the edge weight $\alpha_{ijk} = 1$. When combining all constraints in a graph, the weights α_{ijk} are accumulated. Inconsistencies then appear as cycles in the graph, which in their most common form are of length 2:

$$(i, j) \xrightarrow{\alpha_{ijk}} (i, k) \\ \xleftarrow{\alpha_{ikj}} (i, j)$$

We remedy such cycles by removing the edge with the smaller weight and assigning the weight $|\alpha_{ijk} - \alpha_{ikj}|$ to the remaining edge. For both the MagnaTagATune and CASimIR datasets this already creates a cycle-free graph Q as no larger cycles remain. The cycle-free sets Q are used in this study for training and evaluation.

Compared to the MagnaTagATune dataset, the CASimIR dataset features more frequent recurrences of clips between the triplets presented to the users. Recurring clips relate the corresponding similarity data, and result in large connected components in the CASimIR similarity graph: While the maximal number of clips directly or transitively related to each other through similarity data in the MagnaTagATune dataset was 3 (see [11]), most clips in the CASimIR similarity data are related to at least 5 other clips. The repetition of clips across triplets results in fewer unique referenced clips: the current CASimIR similarity dataset contains only 180 clips referenced by 2102 ratings, while MagnaTagATune references 2000 ratings with about 500 clips, and has 1019 clips with 7650 ratings in total.

3.2 Analysis of Age-bounded Similarity Ratings

The additional participant attributes allow us to select subsets of similarity data according to specific profiles of the participants. This enables the training of more specific models that support better similarity predictions for the relevant group of users, and allows for comparison of different models.

As an example of group-based similarity modelling we choose age as a separating criterion on the CASimIR similarity data from over 256 participants: We divide the complete set of similarity ratings R into two *age-bounded* subsets $R^{\leq 25}$ of data provided by participants not older than 25 years and $R^{> 25}$ containing data of older participants. The boundary of 25 years was chosen as the best approximation to equal sizes of the subsets (data input is only in 5 year bands). As shown in Table 1, the number of ratings is higher for the $R^{\leq 25}$ dataset.

	R	$R^{\leq 25}$	$R^{> 25}$	$R^{\mathbb{b}(\leq 25)}$	$R^{\mathbb{b}(> 25)}$
ratings	2102	919	644	1183	1458
constr.	914	723	576	732	809
clips	180	171	163	175	176

Table 1. Number of votes, unique constraints and referenced clips, after filtering inconsistencies, per dataset.

539 similarity ratings are not associated to a valid age and stored separately in R^0 . For the two age-bounded datasets, we furthermore define complementary datasets $R^{\mathbb{b}(\leq 25)}$ and $R^{\mathbb{b}(> 25)}$ combining the remaining similarity data, e.g. $R^{\mathbb{b}(\leq 25)} = R^{> 25} \cup R^0$. These complementary sets will be used for training of template models for transfer learning.

After splitting, the above (sub)sets of ratings are transferred into constraints (see Section 3.1) and separately filtered for inconsistencies. We now use the corresponding sets of unique constraints Q , $Q^{\leq 25}$, $Q^{> 25}$, $Q^{\mathbb{b}(\leq 25)}$ and $Q^{\mathbb{b}(> 25)}$ for training and testing of models. The number of constraints are also noted in Table 1, together with the total number of clips referenced by the constraint sets. Due to multiple ratings referring to the same constraint and filtering the constraint count is lower than the number of ratings.

4. SIMILARITY MODELLING

The computational representations of music through features, related to physical, musical, and cultural attributes determine the basis of similarity models. Both the Magna-TagATune and CASimIR datasets contain pre-computed features created by The Echo Nest API. For our experiments with CASimIR we derive acoustic features from this data which are aggregated to the clip-level. The 41-dimensional features contain 12 chroma and 12 timbre features, both aggregated via averaging, 2 weight vectors and further features after [8, 11]:

chroma	timbre
segmentDurationMean	tempo
segmentDurationVariance	beatVariance
timeLoudnessMaxMean	tatum
loudness	tatumConfidence
loudnessMaxMean	numTatumsPerBeat
loudnessMaxVariance	timeSignature
loudnessBeginMean	timeSignatureStability
loudnessBeginVariance	–

Table 2. Features used in our experiments.

For experiments with the MagnaTagATune dataset we will use the similar features provided in [12] which contain pre-processed tag information in addition to the acoustic features described above. For the CASimIR dataset, using unprocessed tags from Last.fm did not increase performance in earlier experiments due to very sparse tag assignments. Therefore, our experiments on CASimIR use acoustic features only. For a clip C_i , we refer to its feature vector as $x_i \in \mathbb{R}^N$.

4.1 Mahalanobis Distances

We use the inverse of the distance of two feature vectors as the similarity of the two corresponding clips. The mathematical form of the Mahalanobis distance is used to specify a parametrised distance measure. Given two feature vectors $x_i, x_j \in \mathbb{R}^N$, the distance can be expressed as

$$d_W(x_i, x_j) = \sqrt{(x_i - x_j)^T W (x_i - x_j)},$$

where $W \in \mathbb{R}^{N \times N}$ is a square matrix parametrising the distance function: the *Mahalanobis matrix*. d_W qualifies as a metric if W is positive definite and symmetric.

5. MODEL TRAINING WITH RITML

We now discuss our algorithm which can adapt Mahalanobis distances in order to fit relative similarity data. It is based on the ITML algorithm as described below, which cannot be used directly with relative similarity data. Instead, ITML requires upper or lower bounds on the similarity of two clips, e.g. $d_W(x_i, x_j) < m_{i,j}$ for similar clips. In Section 5.2 we will iteratively derive such constraints during the RITML optimisation process.

5.1 Information-Theoretic Metric Learning

Davis et al. [1] describe Information-Theoretic Metric Learning (ITML) for learning a Mahalanobis distance from absolute distance constraints (e.g. requiring $d_W(x_i, x_j) < 0.5$). A particularly interesting feature of ITML is that a template Mahalanobis matrix $W_0 \in \mathbb{R}^{n \times n}$ can be provided for regularisation. This W_0 can be from a metric that is predefined or learnt on a different dataset. If W_0 is not specified, the identity transform is used. The regularisation of ITML exploits an interpretation of Mahalanobis matrices as multivariate Gaussian distributions: The distance between two Mahalanobis distance functions parametrised by W and W_0 is measured by the relative entropy of the corresponding distributions, which in [1] uses the LogDet divergence D_{ld} :

$$D_{ld}(W, W_0) = \text{tr}(W W_0^{-1}) - \log \det(W W_0^{-1}) - n \\ = 2 * \text{KL}(P(x_i; W_0) \parallel P(x_i; W)).$$

KL refers to the Kullback-Leibler divergence. For details of the transformation see [1]. Given the constraints in form of similar (R_s) and dissimilar (R_d) clip indices as well as upper and lower bounds u_{ij}, l_{ij} , the optimisation problem is then posed as follows:

$$\text{ITML}(W, \xi, c, R_s, R_d) = \\ \underset{W \geq 0, \xi}{\text{argmin}} D_{ld}(W, W_0) + c \cdot D_{ld}(\text{diag}(\xi), \text{diag}(\xi_0)) \\ \text{s.t. } \text{tr}(W d_{i,j}^L (d_{i,j}^L)^T) \leq \xi_{ij} \quad \forall (i, j) \in R_s \\ \text{tr}(W d_{i,j}^U (d_{i,j}^U)^T) \geq \xi_{ij} \quad \forall (i, j) \in R_d \\ \text{with } d_{i,j}^L = (x_i - x_j).$$

Here, ξ_{ij} are slack variables enabling and controlling the violation of individual constraints. The ξ_{ij} are initialised to given upper bounds u_{ij} , if $(i, j) \in R_s$ or lower bounds l_{ij} , if $(i, j) \in R_d$. During optimisation, they are regularised by comparison to the template slack ξ_0 using triangular matrices $\text{diag}(\xi)$ and $\text{diag}(\xi_0)$.

5.2 Relative Learning with RITML

In order to allow for training with relative similarity constraints, we present Relative Information-Theoretic Metric Learning (RITML) based on ITML. Motivated by [14], we embed ITML into an iterative adaptation of the upper and lower bounds.

We start with a training set of relative constraints $(i, j, k) \in Q_t$. We require standard ITML parameters such as c , as well as the relative learning parameters including shrinkage factor η , margin τ and number of cycles k at the beginning. We use the identity matrix for the template W_0 . During iteration m , the active training set of violated constraints Q^m is calculated as

$$Q^m = \{(i, j, k) \in Q_t \mid d_{W^m}(x_i, x_j) > d_{W^m}(x_i, x_k)\}.$$

Q^m is then further divided into the sets of similar and dissimilar constraints R_s^m and R_d^m :

$$R_s^m = \{(i, j) \mid (i, j, k) \in Q^m\}$$

$$R_d^m = \{(i, k) \mid (i, j, k) \in Q^m\},$$

Afterwards, absolute distance constraints ξ_{ij} for the following ITML instance are acquired by adding a margin τ to the average distance values $\mu = \frac{d_{W^m}(x_i, x_j) + d_{W^m}(x_i, x_k)}{2}$ of the clip pairs:

$$\xi_{ij}^m = \begin{cases} \mu - \tau & (i, j) \in R_s^m \\ \mu + \tau & (i, j) \in R_d^m \end{cases} \quad \forall (i, j, k) \in Q^m$$

Now, with ξ^m containing the upper and lower bounds, ΔW can be calculated using

$$\Delta W = \text{ITML}(W^m, \xi^m, \gamma, R_s^m, R_d^m) \quad (1)$$

and the final Mahalanobis matrix is accumulated over iterations using the model update function

$$W^{m+1} = \frac{m * W^m + \eta * \Delta W}{m + 1}.$$

In order for the algorithm to converge, the cardinality of the active training set $|Q^m|$ needs to decrease. In our experiments, $k = 200$ training iterations are usually sufficient. Otherwise an early stopping of the algorithm takes place if $|Q^m|$ does not decrease for 50 iterations. In this case the W^m for the smallest $|Q^m|$ within the last 50 iterations is returned. RITML does not guarantee d_W to be a metric.

Algorithm 1: Relative Training with RITML

Data: Constraints Q_t , features x_i , template matrix W_0 , regularisation factor c , shrinkage factor η , margin τ , number of cycles k

$m = 0$;

while $m \leq k \wedge Q^m \neq \emptyset$ **do**

 Update training sets Q^m , R_s^m and R_d^m ;

 Update absolute constraints ξ^m ;

 Calculate parameter change ΔW ;

 Calculate W^{m+1} ;

$m = m + 1$;

end

return Mahalanobis matrix W^k

5.3 Transfer Learning with W_0 -RITML

The property that motivates our usage of RITML is that it enables *transfer learning*: If a specific starting value or template of W_0 other than the identity matrix is provided, the optimisation tends to produce results close to the provided W_0 . In order to sustain this effect for large numbers of iterations we modify Equation (1) such that regularisation is fixed towards W_0 instead of the Euclidean distance:

$$\Delta W = \text{ITML}(W_0, \xi^m, \gamma, R_s^m, R_d^m)$$

This constitutes the W_0 -RITML algorithm for transfer learning with Mahalanobis matrices.

6. EXPERIMENTS

For all our experiments we use the 10-fold cross-validation with *inductive sampling* as described in [11]: Instead of dividing the similarity constraints themselves into test/training sets, the data are divided on the basis of connected clusters in the similarity data. This approach prevents the recurrence of clips from a training-set in the corresponding test set. It also leads to a greater variance in test-set sizes for CASimIR where the clusters of connected similarity data are larger.

We evaluate the algorithms' performance based on the percentage of training and test constraints fulfilled by the trained model. Our main focus is on the test-set results as we are interested how well the learnt models generalise to unseen data. As a baseline we use the Euclidean distance on the features. We have tested results for statistical significance using the Wilcoxon signed rank test on cross-validation folds' results with a threshold of $p < 5\%$.

Both SVM as implemented in *svmlight*[7] and RITML have hyper-parameters affecting the performance on different datasets. The results reported here were selected on the basis of best test-set performances after a grid-search over a range of value combinations identified as reasonable in preliminary experiments: The regularisation trade-off c is a parameter common to SVM, RITML and W_0 -RITML with a similar effective range: we explored a $c \in [0.001, 10]$ using an approximately logarithmic scale. For RITML and

W_0 -RITML we additionally used
 $\tau \in \{10^{-4}, 10^{-3}, \dots, 10^{-1}, 0.5, 1, \dots, 10\}$ and
 $\eta \in \{0.1, 0.15, \dots, 0.95\}$.

6.1 Comparing the Performance of RITML

For a comparable evaluation of RITML we chose the MagnaTagATune-based dataset and constraint sampling published in [12]. Their evaluation compares various algorithms for learning a Mahalanobis metric using two different samplings. The inductive sampling used here corresponds to the *sampling B* in their text. Table 3 shows the results on MagnaTagATune and on the complete CASimIR dataset (Q).

Algorithm	MagnaTagATune	CASimIR
Euclidean	59.80 / 59.77	59.75 / 59.82
RITML	71.12 / 73.41	64.23 / 93.36
SVM	71.20 / 85.75	63.22 / 69.11
MLR	68.90 / 100.0	62.79 / 73.37

Table 3. Comparison of Test / Training set performance on the MagnaTagATune and CASimIR datasets for baseline, RITML and SVM. Reported are the number of constraints fulfilled by the learnt distance measures.

For MagnaTagATune, RITML achieves similar generalisation results as SVM (with parameters SVM: $c = 0.7$ and RITML: $c = 1, \eta = 0.85, \tau = 0.5$), while MLR overfits to the training data. For both the MagnaTagATune and CASimIR datasets all methods perform significantly better than the baseline. The RITML results are therefore comparable to the state-of-the-art. The training results on MagnaTagATune with SVM and MLR are far better than the test results, indicating overfitting, which does not occur for RITML. Interestingly, on the CASimIR dataset, the situation between RITML and SVM is reversed. Results published by [11] for acoustic-only features on MagnaTagATune show a performance of 66% on MagnaTagATune, but the lower performance on CASimIR can be explained by the smaller number of training examples.

6.2 Transfer Learning

A core motivation for transfer learning is the training on highly specialised but small datasets. To evaluate the W_0 -RITML method for transfer learning, we firstly compared the SVM and RITML algorithms with the baseline on the age-bounded datasets $Q^{>25}$ and $Q^{\leq 25}$ in Table 4. The rightmost column shows the average performance across both age-bounded datasets. Expectedly, on these smaller datasets generalisation results for RITML as well as the reference SVM and MLR are lower than on the whole CASimIR. Only for RITML an increase of 4.37% from the baseline is notable for the slightly larger $Q^{>25}$ which improves the average score for RITML.

We now apply transfer learning to improve generalisation results on the age-bounded sets. The overall process is depicted in Figure 1. First, a similarity modelling experiment is performed on both of the complementary subsets

Algorithm	$Q^{\leq 25}$	$Q^{>25}$	Average
Euclidean	59.32 / 59.95	59.15 / 59.63	59.23 / 59.79
RITML	63.69 / 75.87	61.02 / 67.95	62.35 / 71.91
SVM	61.56 / 72.78	61.34 / 71.43	61.45 / 72.10
MLR	62.06 / 75.79	62.58 / 78.47	62.32 / 77.13
W_0 -Direct	63.96 / 66.17	64.82 / 69.57	64.39 / 67.87
W_0 -RITML	65.53 / 70.82	67.07 / 73.22	66.30 / 72.02

Table 4. Comparison of Test / Training set performance on the age-bounded datasets. Training on single datasets (top 3 rows) and transfer learning with W_0 -RITML and W_0 -Direct.

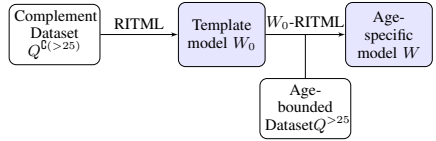


Figure 1. Flow diagram for transfer learning, exemplified for the $Q^{>25}$ dataset.

$Q^{b(>25)}$ and $Q^{b(\leq 25)}$ using cross-validation with training and test data from only these sets. Comparing the individual results for validation folds we choose the Mahalanobis matrices with the greatest test-set performance as template matrix W_0 . The template matrix W_0 learnt on $Q^{b(\leq 25)}$ is then used for transfer learning on $Q^{\leq 25}$, using W_0 -RITML. For comparison of the effectiveness of the fine-tuning with W_0 -RITML, we report the performance achieved with the unmodified W_0 on $Q^{\leq 25}$ as W_0 -Direct. This process is repeated analogously for $Q^{\leq 25}$ by applying the template matrix W_0 from $Q^{b(\leq 25)}$ on $Q^{>25}$.

The highlighted lower columns of Table 4 show the results for transfer learning: Row W_0 -Direct reports the direct performances of the template Mahalanobis matrices W_0 . The results of fine-tuning these models with W_0 -RITML are reported in the last row. We here find that using the matrices trained on the larger datasets, and thus transfer learning, generally improves results. Only the results for W_0 -RITML provide gains $> 6.21\%$ that are statistically significant when compared to the baseline. As the average result of W_0 -RITML also significantly outperforms the average SVM performance, W_0 -RITML works best for adapting models to specialised datasets.

A drawback of RITML is that it is computationally demanding: For the Q dataset, RITML uses 50 seconds where SVM converges in 5 seconds. On the other hand, SVM learns a diagonal W which reduces the number of parameters and model flexibility.

6.3 Model Comparison

In order to identify specificities of the $Q^{>25}$ dataset in comparison to the remaining $Q^{b(>25)}$, we now analyse changes made to the template matrix W_0 in the fine-tuning process. Instead of starting from the Euclidean metric, models learnt from the W_0 -RITML method have a model already

adapted to similarity data as basis.

Figure 2 shows the relative difference $\hat{W} - \hat{W}_0$ of the Mahalanobis matrix before (W_0) and after (W) fine tuning. As the fine tuning process rescales the similarity measure and thereby W , the matrices have been normalised to the interval of $[0, 1]$ via³

$$\hat{W} = \frac{W - \min_{i,j}(w_{ij})}{\max_{i,j}(W - \min_{i,j} w_{ij})_{ij}}. \quad (2)$$

The axes of the figure correspond to feature types, which for better overview have been grouped into chroma, timbre and ranges of the features in Table 2. The template matrix W_0 in Figure 3a has large values only in the diagonal and homogeneous small values off the diagonal. In comparison to this, Figure 2 shows that specific combinations of timbre features (in the bottom centre) with (B)eat and tempo statistics were raised in importance by W_0 -RITML, resulting in the final matrix W as shown in Figure 3b. Also, the centre of the matrix shows increased values for combinations of different timbre coefficients. The strongest increases (20-24%) in weights are reported for the off-diagonal fields of $C_{11}C_1, T_6T_5, B_4T_4$ and B_4T_5 , where C, T relate to chroma and timbre coefficients and B_4 refers to the tatumConfidence feature. Weights are increased mainly at the cost of diagonal elements, and suggest at a specialisation of the model to the specificities of the $Q^{>25}$ similarity subset. For this data collected from users aged over 25, the analysed W_0 -RITML model with stronger influence of the timbre and beat-statistics features performs best in our evaluation.

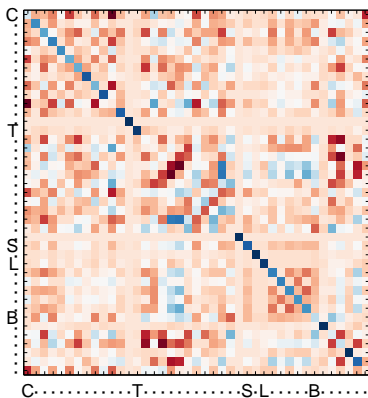


Figure 2. Learnt model difference for W_0 -RITML on $Q^{>25}$. Axis labels represent ranges of feature types: (C)hroma, (T)imbre, as well as (S)egment, (L)oudness and (B)eat+Tempo statistics. Dark red / blue colours correspond to strong weight increase / decrease.

³ Subtraction and division are applied to W in a point-wise manner.

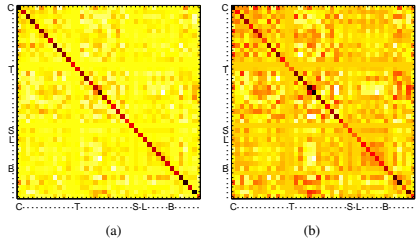


Figure 3. (a) Template matrix W_0 before and (b) final matrix W after fine-tuning with W_0 -RITML on $Q^{>25}$. The latter shows higher variance in off-diagonal entries for the specialised model. Axis labels represent ranges of feature types: (C)hroma, (T)imbre, as well as (S)egment, (L)oudness and (B)eat+Tempo statistics. Dark red colours correspond to strong weight increase, light yellow to decrease.

7. CONCLUSION & FUTURE WORK

We presented a method for analysing music similarity data of different user groups via models trained with transfer learning. To this end, the new RITML algorithm was developed extending ITML to relative similarity data. A key feature of RITML is that it enables transfer learning with template Mahalanobis matrices via W_0 -RITML. Our evaluation of the algorithm was performed on two datasets: The evaluation on the commonly used MagnaTagATune dataset showed that RITML performs comparably to state-of-the-art algorithms for metric learning.

For evaluation of transfer learning with W_0 -RITML we provide the CASimIR similarity dataset, the first open dataset containing user attributes associated to relative similarity data. Tests on the whole CASimIR dataset corroborated our finding that RITML competes with current similarity learning methods. Our analysis of W_0 -RITML was performed on age-bounded subsets of the dataset. Results showed that transfer learning with W_0 -RITML outperforms the standard SVM algorithm on small datasets.

Our comparison of models allowed us to point out specific features and combinations that determine similarity in user data. For this first evaluation we chose age to group users. We hope this will motivate further research in comparison of similarity models and adaptation to data with regard to cultural and user context.

For future work we are interested in collecting larger similarity datasets, and applying the methods introduced here for improved validation of results and the analysis of more specific user groups. The set-up used for our experiments motivates transfer learning across the MagnaTagATune and CASimIR datasets with W_0 -RITML for further analysis of the transferability of similarity information via Mahalanobis matrices.

8. REFERENCES

- [1] Jason V. Davis, B. Kulis, Prateek Jain, Suvrit Sra, and Inderjit S. Dhillon. Information-theoretic metric learning. In *Proc. of ICML '07*, pages 209–216, New York, NY, USA, 2007. ACM.
- [2] Philippe Hamel, Matthew E. P. Davies, Kazuyoshi Yoshii, and Masataka Goto. Transfer learning in mir: Sharing learned latent representations for music audio classification and similarity. In Alceu de Souza Britto Jr., Fabien Gouyon, and Simon Dixon, editors, *ISMIR*, pages 9–14, 2013.
- [3] Edith Law and Luis Von Ahn. Input-agreement: A new mechanism for collecting data using human computation games. In *Proc. of CHI*. ACM Press, 2009.
- [4] B. McFee, L. Barrington, and G. Lanckriet. Learning similarity from collaborative filters. In *Proc. of ISMIR 2010*, pages 345–350, 2010.
- [5] Brian McFee and Gert R. G. Lanckriet. Partial order embedding with multiple kernels. In *Proc. of the 26th International Conference on Machine Learning (ICML'09)*, pages 721–728, June 2009.
- [6] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, October 2010.
- [7] M. Schultz and T. Joachims. Learning a distance metric from relative comparisons. In *Advances in Neural Information Processing Systems (NIPS)*. MIT Press, 2003.
- [8] Malcolm Slaney, Kilian Q. Weinberger, and William White. Learning a metric for music similarity. In Juan Pablo Bello, Elaine Chew, and Douglas Turnbull, editors, *Proc. of ISMIR 2008*, pages 313–318, 2008.
- [9] Sebastian Stober. *Adaptive Methods for User-Centered Organization of Music Collections*. PhD thesis, Otto-von-Guericke-University, Magdeburg, Germany, Nov 2011. published by Dr. Hut Verlag, ISBN 978-3-8439-0229-8.
- [10] Sebastian Stober and Andreas Nürnberger. Similarity adaptation in an exploratory retrieval scenario. In *Proc. of AMR 2010*, Linz, Austria, Aug 2010.
- [11] Daniel Wolff and Tillman Weyde. Learning music similarity from relative user ratings. *Information Retrieval*, pages 1–28, 2013.
- [12] Daniel Wolff, Sebastian Stober, Andreas Nürnberger, and Tillman Weyde. A systematic comparison of music similarity adaptation approaches. In *Proc. of ISMIR 2012*, pages 103–108, 2012.
- [13] Daniel Wolff, Guillaume Bellec, Anders Friberg, Andrew MacFarlane, and Tillman Weyde. Creating audio based experiments as social web games with the casimir framework. In *Proc. of AES 53rd International Conference: Semantic Audio*, Jan 2014.
- [14] Zhaohui Zheng, Keke Chen, Gordon Sun, and Hongyuan Zha. A regression framework for learning ranking functions using relative relevance judgments. In *Proc. of SIGIR '07*, pages 287–294, New York, NY, USA, 2007. ACM.

MODELING GENRE WITH THE MUSIC GENOME PROJECT: COMPARING HUMAN-LABELED ATTRIBUTES AND AUDIO FEATURES

Matthew Prockup⁺⁺, Andreas F. Ehmann*, Fabien Gouyon*

Erik M. Schmidt*, Oscar Celma*, and Youngmoo E. Kim⁺

Electrical and Computer Engineering, Drexel University⁺ and Pandora Media Inc.*

{mprockup, ykim}@drexel.edu {aehmann, fgouyon, eschmidt, ocelma}@pandora.com

ABSTRACT

Genre provides one of the most convenient categorizations of music, but it is often regarded as a poorly defined or largely subjective musical construct. In this work, we provide evidence that musical genres can to a large extent be objectively modeled via a combination of musical attributes. We employ a data-driven approach utilizing a subset of 48 hand-labeled musical attributes comprising instrumentation, timbre, and rhythm across more than one million examples from Pandora[®] Internet Radio's *Music Genome Project*[®]. A set of audio features motivated by timbre and rhythm are then implemented to model genre both directly and through audio-driven models derived from the hand-labeled musical attributes. In most cases, machine learning models built directly from hand-labeled attributes outperform models based on audio features. Among the audio-based models, those that combine audio features and learned musical attributes perform better than those derived from audio features alone.

1. INTRODUCTION

Musical *genre* is a high-level label given to a piece of music (e.g., Rock, Jazz) to both associate it with similar music pieces and distinguish it from others. Genre is a very popular way to organize music as it is being used by virtually all actors in the music industry, from record labels and music retailers, to music consumers and musicians via radio and music streaming services on the internet.

Just because genres are widely used does not necessarily mean that they are easy to categorize, or easy to recognize. In fact, previous research shows that the music industry uses inconsistent genre taxonomies [21], and there is debate over whether genre is the product of objective or subjective categorizations [28]. Furthermore, it is debated whether individual musical properties (e.g. tempo, rhythm, instrumentation), which are not always exclusive to a sin-

gle genre, represent defining components [1, 10]. For example, an Afro-Latin clave pattern occurs many places, both in Antonio Carlos Jobim's *The Girl from Ipanema* (Jazz) and in The Beatles' *And I Love Her* (Rock). It can even be heard in the recently popular song, *All About that Bass*, by Meghan Trainor. However, when discriminating the more specific subgenres of 'Bebop' Jazz (fast swing) and 'Brazilian' Jazz (Afro-Latin rhythms), this clave property becomes much more salient. Despite these intriguing relationships, a large-scale analysis of the association of musical properties to genre, to the knowledge of the authors, has yet to be performed.

If it were possible to define a categorization of music genres that is useful, meaningful, consensual and consistent *at some level*, then an automated categorization of music pieces into genres would be both achievable and highly desirable. Since early research in Music Information Retrieval (MIR), and still to date, the automatic genre recognition from music pieces has precisely been an important topic [1, 28, 30].

In this work, we explore the intriguing relationship of genre and musical attributes. In Section 3, we will overview the expertly-curated data used. In Section 4, we detail an applied musicology experiment that uses expertly-labeled musical attributes to model genre. We then report in Section 5 on a series of experiments regarding automated categorization of music pieces into genres using audio signal analysis. In the following section, we will briefly outline each of these approaches.

2. APPROACH

In this work we explore four approaches to modeling musical genre, investigating both expert human annotations as well as audio representations (Figure 1). We explore a subset of 12 'Basic' musical genres (e.g. Jazz) as well as a selected subset of 47 subgenres (e.g. Bebop). In the first approach, we address via data-driven experiments whether objective musical attributes of music pieces carry sufficient information to categorize their genre. The next set of approaches uses audio features to model genre automatically. In the second approach, we use audio features directly. The third approach uses audio features to model each of the musical attributes individually, which are then used to model genre. In the fourth approach, the estimated attributes are used in conjunction with raw audio features.



© Matthew Prockup, Andreas F. Ehmann, Fabien Gouyon Erik M. Schmidt, Oscar Celma, and Youngmoo E. Kim. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Matthew Prockup, Andreas F. Ehmann, Fabien Gouyon Erik M. Schmidt, Oscar Celma, and Youngmoo E. Kim. "Modeling Genre with the Music Genome Project: Comparing Human-Labeled Attributes and Audio Features", 16th International Society for Music Information Retrieval Conference, 2015.

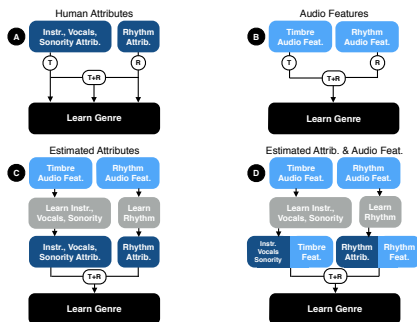


Figure 1. An overview of the experiments performed.

By injecting human-inspired context, we hope to automatically capture elements of genre in a manner similar to that of models derived from attributes labeled by music experts.

3. DATA - THE MUSIC GENOME PROJECT®

Both the musical attribute and genre labels used were defined and collected by musical experts on a corpus of over one million music pieces from Pandora® Internet Radio's *Music Genome Project*® (MGP)¹. The labels were collected over a period of nearly 15 years and great care was placed in defining them and analyzing each song with that consistent set of criteria.

3.1 Musical Attributes

The musical attributes refer to specific musical components comprising elements of the vocals, instrumentation, sonority, and rhythm. They are designed to have a generalized meaning across all genres (in western music) and map to specific and deterministic musical qualities. In this work, we choose subset of 48 attributes (10 rhythm, 38 timbre). An overview of the attributes is shown in Table 1.

Meter attributes denote musical meters separate from simple duple (e.g. cut-time, compound-duple, odd)

Rhythmic Feel attributes denote rhythmic interpretation (e.g., swing, shuffle, back-beat strength) and elements of rhythmic perception (e.g., syncopation, danceability)

Vocal attributes denote the presence of vocals and timbral characteristics of voice (e.g., male, female, vocal grittiness).

Instrumentation attributes denote the presence of instruments (e.g., piano) and their timbre (e.g., guitar distortion)

Sonority attributes describe production techniques (e.g., studio, live) and the overall sound (e.g., acoustic, synthesized)

Table 1. Explanations of rhythm and timbre attributes.

¹ "Pandora" and "Music Genome Project" are registered trademarks of Pandora Media, Inc. <http://www.pandora.com/about/mgp>

Each of the attributes is rated on continuous scale from 0-1. In some contexts, it is helpful to convert them to binary labels if they show only low (absence) or high (presence) ratings with little in between [25].

3.2 Genre and Subgenre

In this work we will explore a selected subset of 12 'Basic' genres and 47 additional sub-genres. 'Basic' genre is assembled as a mix of very expansive genres (e.g., Rock, Jazz) as well as some more focused ones (e.g., Disco and Bluegrass), serving as an analog to many previous genre experiments in MIR. The presence of a genre is notated independently for each song by a binary label. A selection of genre labels and a simplistic high-level organization for discussion purposes is shown in Table 2.

Basic Genre: Rock, Jazz, Rap, Latin, Disco, Bluegrass, etc.

Jazz Subgenre: Cool, Fusion, Hard Bop, Afro-Cuban, etc.

Rock Subgenre: Light, Hard, Punk, etc.

Rap Subgenre: Party, Old School, Hardcore, etc.

Dance Subgenre: Trance, House, etc.

World Subgenre: Cajun, North African, Indian, Celtic, etc.

Table 2. Some of the musical genres and subgenres used.

4. MUSICAL ATTRIBUTE MODELS OF GENRE

In order to see the extent to which genre can be modeled by musical attributes, we first perform an applied musicology experiment using the set of expertly-labeled attributes from Section 3.1 and relate them to labels of genre. A model for each individual genre is trained on each of the musical attributes alone and in rhythm- and timbre-based aggregations. This will show the role that each attribute or collection of attributes plays and how they interact with one another in order to create joint representations of genre. Each model employs logistic regression trained using stochastic gradient descent (SGD) [25]. The training data was separated on a randomly shuffled 70%:30% (train:test) split with no shared artists between training and testing. Due to the size of the dataset, a single trial for each attribute is both tractable and sufficient. The learning rate for each genre model is tuned adaptively.

4.1 Evaluating the Role of Musical Attributes

In order to evaluate each of the models, the area under the receiver operating characteristic (ROC) curve will be used. Each genre has large and varying class imbalance, so this is first corrected for by weighting training examples appropriately in the cost function. However, accuracy alone still does not tell the whole story. High accuracy can be achieved by predicting only the negative class (genre absence). Area under the ROC curve allows for a more comparable difference between each of the models than raw accuracy alone. It gives insight into the trade-off between true positive and false positive rates. Alternatively

we could have used precision and recall (PR) curves for evaluation, but it is shown that if one model dominates in the ROC domain, it will also dominate in the PR domain and vice-versa [5]. In this work, the area under the ROC curve will be referred to as AUC.

The results for each of the attribute-based genre models are shown in Tables 3 and 4. The tables outline the AUC values for classifying genre using timbre attributes, rhythm attributes, and their combination. Table 3 summarizes all results, showing the mean of all AUC values for each genre model contained in the subgroups defined in Section 3.2. Using attributes of rhythm and timbre together show better performance than using each alone. Secondly, timbre tends to perform better than rhythm. This suggests that the timbre attributes in this context are better descriptors. However in some cases, the rhythm attributes, even though there are less of them (10 rhythm, 38 timbre), are not that far behind. They are especially important in defining Jazz and Rap, where rhythms such as swing in Jazz or syncopated vocal cadences over back-beat heavy drums in Rap play defining roles.

Genre Group	Timbre	Rhythm	Both
Basic	0.905	0.841	0.918
Rock Sub	0.910	0.819	0.919
Jazz Sub	0.925	0.856	0.945
Rap Sub	0.901	0.891	0.940
Dance Sub	0.961	0.881	0.965
World Sub	0.885	0.833	0.904
Mean	0.913	0.848	0.931

Table 3. An overview of all models using musical attributes.

In Table 4 we show the individual AUC results for the set of ‘Basic’ genres and subgenres of Jazz. Within these individual groups, rhythm and timbre attributes together are once again able to better represent genre than when used individually. Each of the ‘Basic’ genres can be represented reasonably well with just timbre, as each has slightly differing instrumentation. However, we again see the importance of rhythm, describing what instrumentation and timbre cannot capture alone. Genres heavily reliant on specific rhythms (e.g., Funk, Rap, Latin, Disco, Jazz) are all able to be represented rather well with only rhythm attributes. In the Jazz subgenre this emphasis on rhythm in certain cases is even more clear. In the next subsection, we will dive deeper into the attributes that best describe the Jazz subgenres.

4.2 The Influence of Rhythm and Timbre in Jazz

In order to more deeply explore the defining relationships of rhythm and instrumentation within a subgenre, we will look further into Jazz. Table 5 shows a subset of the important musical attributes for the Jazz subgenres. The AUC accuracy of classifying each subgenre based on individual musical attributes is shown.

The presence of solo brass (e.g., trumpet), piano, reeds (e.g., saxophone) and auxiliary percussion (e.g., congas) are important defining characteristics of instrumentation.

Basic Genre	Timbre			Jazz Subgenre			
	Timbre	Rhythm	Both	Timbre	Rhythm	Both	
Rock	0.843	0.759	0.856	New Orleans	0.970	0.957	0.989
Blues	0.913	0.783	0.915	Boogie	0.943	0.893	0.978
Gospel	0.810	0.664	0.843	Swing	0.970	0.933	0.984
Soul	0.869	0.793	0.887	Bebop	0.976	0.965	0.988
Funk	0.937	0.862	0.937	Cool	0.964	0.928	0.975
Rap	0.926	0.890	0.951	Hard Bop	0.944	0.905	0.967
Folk	0.943	0.760	0.952	Fusion	0.843	0.750	0.886
Country	0.952	0.794	0.955	Free	0.906	0.855	0.936
Reggae	0.893	0.819	0.905	Afro-Cuban	0.961	0.910	0.972
Latin	0.940	0.904	0.945	Brazilian	0.871	0.847	0.905
Disco	0.899	0.891	0.902	Acid	0.886	0.660	0.891
Jazz	0.937	0.850	0.963	Smooth	0.862	0.667	0.871
Mean	0.905	0.814	0.918	Mean	0.925	0.856	0.945

Table 4. Experimental results for ‘Basic’ genre and Jazz subgenre models using musical attributes.

Jazz Subgenre	Timbre					Rhythm				
	Solo Brass	Piano	Reeds	Aux. Perc.	BackBeat	Dance	Swing	Shuffle	Syncop.	
New Orleans	0.888	0.786	0.790	0.680	0.652	0.564	0.936*	0.513	0.513	
Boogie	0.511	0.721	0.784	0.778	0.592	0.727	0.737	0.508	0.984	
Swing	0.721	0.784	0.748	0.679	0.624	0.578	0.923*	0.511	0.508	
Bebop	0.725	0.850	0.862	0.703	0.662	0.525	0.946*	0.509	0.602	
Cool	0.639	0.750	0.836	0.701	0.697	0.424	0.890*	0.504	0.568	
HardBop	0.606	0.774	0.737	0.669	0.726	0.555	0.808*	0.684	0.606	
Fusion	0.604	0.497	0.669	0.507	0.574	0.577	0.507	0.500	0.693*	
Free	0.606	0.338	0.784	0.615	0.809*	0.765	0.577	0.515	0.558	
Afro-Cuban	0.696	0.822	0.786	0.832*	0.782	0.648	0.512	0.501	0.790	
Brazilian	0.560	0.736	0.568	0.572	0.761*	0.555	0.532	0.504	0.635	
Acid	0.591	0.513	0.658*	0.507	0.585	0.622	0.509	0.515	0.635	
Smooth	0.530	0.577	0.748*	0.590	0.559	0.614	0.513	0.509	0.573	

Table 5. Attributes important to the Jazz subgenres are shown. AUC values greater than 0.70 are bold. The highest performing attribute for each genre is denoted with a *.

Boogie and Afro-Cuban styles, even though different, place heavy emphasis on the piano, which is shown here as well. Bebop, Hard-bop, and Afro-Cuban Jazz show emphasis placed on solo brass, piano, and reeds, as they rely heavily on solo artists of these instruments (e.g., ‘Dizzy’ Gillespie, Miles Davis, Thelonious Monk, John Coltrane). The presence of auxiliary percussion is also a good descriptor of Afro-Cuban Jazz, where the use of hand drums (e.g., bongos, congas) is very prevalent.

Rhythm is also important in Jazz subgenres. The danceability, back-beat, and presence of swing and syncopation are defining characteristics of certain Jazz rhythms. It is important to note that a high AUC does not necessarily denote the presence of that attribute, only its consistent relationship. For example, back-beat is a good predictor of Free Jazz possibly due to its absolute absence. Alternatively, one may think that the presence of swing is important in all Jazz. Bebop, Hard Bop, New Orleans, and Swing Jazz do have a heavy dependence on swing being present. However, Afro-Cuban Jazz relies on straight time, clave-based rhythms, so syncopation is actually a better predictor. It is also important to note that while the attributes of swing and shuffle are musically related, there is a clear distinction in their application. In this case, swing is very important, while shuffle is only slightly useful (e.g., Boogie). However, outside of the Jazz genre, the opposite case may be true, where shuffle is the more important attribute (e.g. Blues, Country). This suggests that it is important to make a clear distinction between swing and shuffle.

5. PREDICTING GENRE FROM AUDIO

There is a large body of work on musical genre recognition from and audio signals [28,30]. However, most known prior work in this area focuses on discriminating a discrete set of basic genre labels with little emphasis on what defines genre. In response, researchers have tried to develop datasets that focus on style or subgenre labels (e.g., ballroom dance [7, 13, 24], latin [19], electronic dance [23], Indian [17]) that have clear relations to the presence of specific musical attributes. However, because models are designed for these specific sets, the methods used do not adapt to larger more generalized music collections. For example, tempo alone is a good descriptor for the ballroom dance style dataset, which is not true for more general collections [12].

Other work in genre recognition avoids the problem of strict genre class separations. Audio feature similarity, self organizing maps, and nearest-neighbor approaches can be used estimate genre of an unknown example [22]. Similarly, auto-tagging approaches use audio features to learn the presence of both musical attributes and genre tags curated by the public [2, 8] or by experts [29].

In this work, we compare modeling genre both with audio features directly and with stacked approaches that exploit the relationships of audio features and musical attributes.

5.1 Timbre Related Features

In order to capture timbral components and model vocal, instrumentation, and sonority attributes, block-based Mel-Frequency Cepstral Coefficients (MFCC) are implemented. Means and covariances of 20 MFCCs are calculated across non-overlapping 3-second blocks. These block-covariances are further summarized over the piece by calculating their means and variances [27]. This yields a 460 dimensional timbre based feature set.

5.2 Rhythm Related Features

In order to capture aspects of each rhythm attribute, a set of rhythm-specific features was implemented. All rhythm features described in this section rely on global estimates of an accent signal [3]

The *beat profile* quantizes the accent signal between consecutive beats to 36 subdivisions. The beat profile features are statistics of those 36 bins over all beats. The feature relies on estimates of both beats [9] and tempo.

The *tempogram ratio* feature (TGR) uses the tempo estimate to remove the tempo dependence in a tempogram. By normalizing the tempo axis of the tempogram by the tempo estimate, a fractional relationship to the tempo is gained. A compact, tempo-invariant feature is created by capturing the weights of the tempogram at musically related ratios relative to the tempo estimate.

The *Mellin scale transform* is a scale invariant transform of a time domain signal. Similar musical patterns at different tempos are scaled relative to the tempo. The Mellin scale transform is invariant to that tempo scaling. It

was first introduced in the context of rhythmic similarity by Holzapfel [16], around which our implementation is based. In order to exploit the natural periodicity in the transform, the discrete cosine transform (DCT) is computed. Median removal (by subtracting the local median) and half-wave rectifying the DCT creates a new feature that emphasizes transform periodicities.

The previous rhythm features are also extended to multiple-band versions by using accent signals that are constrained to be within a set of specific sub-bands. This affords the ability to capture the rhythmic function of instruments in different frequency ranges. The rhythm feature set used in this work is an aggregation of the median removed Mellin Transform DCT and multi-band representations of the beat profile and the tempogram ratio features. This yields a 372 dimensional rhythm based feature set that was shown in previous work to be relatively effective at capturing musical attributes related to rhythm (see [25] for more details).

5.3 Genre Recognition Experiments

In addition to the experiment from Section 4, we present three additional methods for modeling genre, each based on audio signal analysis. The second method (Figure 1b) performs the task of genre recognition with rhythm and timbre inspired audio features directly. The third method (Figure 1c) is motivated similar to the first experiment, which employs the expertly-labeled musical attributes. However, inspired by work in transfer learning [4], audio features are used to develop models for the humanly-defined attributes which in turn are used to model genre. Through this supervised pre-training of musical attributes, models of genre can be learned from attributes' estimated presence. In the fourth approach (Figure 1d), inspired by [6] and [18], the learned attributes are combined with the audio features directly in a shared middle layer to train models of genre.

Similar to Section 4, genre is modeled with logistic regression fit using stochastic gradient descent (SGD). The data was separated on the same 70%:30% (train:test) split. Once again, there were no shared artists between training and testing. Due to the size of the dataset, a single trial for each genre, as well as for each learned musical attribute, is both tractable and sufficient. The learning rate for each model is tuned adaptively.

5.3.1 Using Audio Features Directly

Of the four presented approaches, the second uses audio features directly to model genre. The features from Sections 5.1 and 5.2 are used in aggregation and a model is trained and tested for each individual genre. This provides a baseline for what audio features are able to capture without any added context. However, this lack of context makes it hard to interpret what about genre they are capturing.

5.3.2 Stacked Methods

The third and fourth approaches are also driven by audio features. However instead of targeting genre directly,

models are learned for each of the vocal, instrumentation, sonority, and rhythm attributes. Inspired by approaches in transfer learning [4], and similar in structure to previous methods in the MIR community [20], the learned attributes are then used to predict genre. This approach is formulated similar to a basic neural network with a supervised pre-trained (and no longer hidden) musical attributes layer.

The rhythm-based attributes are modeled with a feature aggregation of the Mellin DCT, multi-band beat profile, and multi-band tempogram ratio features. The vocals, instrumentation, and sonority attributes are modeled with the block-based MFCC features. Each attribute is modeled using logistic regression for binary labels (categorical) and linear regression for continuous labels (scale-based). If an individual attribute is formulated as a binary classification task (see Section 3.1), the probability of the positive class (its presence) is used as the feature value.

The first version of the stacked methods (third approach) uses audio features to estimate musical attributes and employs only those estimated attributes to model genre. The second version (fourth approach) concatenates the audio features and the learned attributes in a shared middle layer to model genre [6, 18].

5.4 Results

In this section, we will give an overview of all of the results from the audio-based methods, and compare them to the models learned from the expertly-labeled attributes. In order to show the overall performance of each method in a compact way, only combined rhythm and timbre approaches will be compared. Once again each genre model will be evaluated using area under the ROC curve (AUC). In order to better evaluate the stacked models, we will finish with a brief evaluation of the learned attributes.

5.4.1 Learning Genre

A summary of the results for the audio experiments using rhythm and timbre features is shown in Table 6. The human attribute model results are also included for comparison. Similar to Table 3, the mean AUC of each genre grouping is shown.

Genre Group	Human Attrib.	Audio Feat.	Learned Attrib.	Audio + Learned
Basic	0.918	0.892	0.878	0.899
Rock Sub	0.919	0.902	0.903	0.911
Jazz Sub	0.945	0.910	0.893	0.923
Rap Sub	0.940	0.916	0.914	0.927
Dance Sub	0.965	0.963	0.955	0.966
World Sub	0.904	0.850	0.846	0.865
Mean	0.931	0.905	0.897	0.915

Table 6. An overview of experimental results using audio-based models that utilize timbre and rhythm features.

Compared to the human attributes approach, using audio features alone to model genre performs relatively well. This is especially true for the ‘Basic’, Rock, and Dance groups, where the audio feature AUC results are very close to human attribute performance. Across the other groups,

the differences between the audio feature models and the musical attribute models suggest that the audio features lose some important, genre-defining information. Furthermore, performance that was close to musical attributes when using only audio features alone is also close when musical attributes learned from audio features. This suggests that, in these cases, the audio features may be capturing similarly salient components related to the musical attributes that describe these genre groups.

Overall, the learned attributes perform just as good as or worse than the audio features alone. This suggests that they are at most as powerful as the audio features used to train them. However, combining audio features and learned attributes shows significant improvement (paired t-test $p < 0.01$ across all genres) over using audio features or learned attributes alone. This evidence suggests that audio features and learned attribute models each contain slightly different information. The added human context of the learned attributes is helpful to achieve results that approach those of the expertly-labeled attributes. This also suggests that the decisions made from learned labels are possibly more similar to the decisions made from human attribute labels, and the errors in estimating the musical attributes are possibly to blame for the performance decrease when used alone.

Basic Genre	Human Attrib.	Audio Feat.	Learned Attrib.	Audio + Learned	Jazz Subgenre	Human Attrib.	Audio Feat.	Learned Attrib.	Audio + Learned
Rock	0.856	0.831	0.835	0.837	New Orleans	0.989	0.987	0.991	0.996
Blues	0.915	0.892	0.883	0.899	Boogie	0.978	0.962	0.939	0.962
Gospel	0.843	0.798	0.794	0.805	Swing	0.984	0.929	0.929	0.940
Soul	0.887	0.833	0.818	0.842	Bluesop	0.988	0.951	0.943	0.957
Funk	0.937	0.911	0.886	0.918	Cool	0.975	0.900	0.901	0.916
Rap	0.951	0.963	0.951	0.969	HardBop	0.967	0.946	0.930	0.952
Folk	0.952	0.905	0.903	0.916	Fusion	0.886	0.844	0.812	0.867
Country	0.955	0.885	0.880	0.897	Free	0.936	0.920	0.923	0.931
Reggae	0.905	0.926	0.885	0.929	AfroCuban	0.972	0.934	0.912	0.946
Latin	0.945	0.921	0.905	0.923	Brazilian	0.905	0.879	0.858	0.904
Disco	0.902	0.936	0.903	0.938	Acid	0.891	0.841	0.763	0.846
Jazz	0.963	0.907	0.906	0.916	Smooth	0.871	0.868	0.853	0.894
Mean	0.918	0.892	0.878	0.899	Mean	0.945	0.910	0.893	0.923

Table 7. Experimental results for the ‘Basic’ genres and Jazz subgenres using audio-based models.

The left half of Table 7 shows the results for predicting the ‘Basic’ genre labels. Within this set, we see some interesting patterns start to emerge. In the case of Rap, Reggae, and Disco, audio features are actually able to outperform the musical attributes. This suggests that our small selected subset of 48 human attribute labels do not always tell the whole story, and that the audio features, which are much larger in dimensionality, possibly contain additional and/or different information. As in previous results, the learned attribute models perform similarly to methods that use audio features directly, but with a few exceptions. In the cases that the audio feature models do better than the human-labeled musical attribute models, the learned attribute models are able to perform *at most* as good as the human-labeled musical attribute models. This once again suggests that the learned attribute approach may be better approximating the decisions the human-labeled attribute approach is making. When adding audio and learned attributes together, the added context is once again beneficial, with combined methods outperforming models that use audio or learned attributes alone. Audio feature models that perform better than the human attributes models

are additionally improved, showing again that the audio features and human attribute labels contain complementary information.

The right half of Table 7 shows the results for predicting the Jazz subgenre labels. The Jazz genre shows more expected relationships between the human attribute, audio feature, and learned attribute methods. The combined method outperforms each of the audio feature and learned attribute methods. The human attribute method performs better than all audio-based methods.

5.4.2 Learning Attributes

In order to further explore the stacked audio-based models, we performed a small evaluation of how well the audio features are able to learn each of the expertly-labeled musical attributes. Sticking with a common theme, we will explore the results of modeling attributes that are important to Jazz (from Table 5). Table 8 shows the ability to directly predict these attributes from audio features. AUC accuracies are reported for the binary attributes; R^2 values are reported for continuous attributes. The results of evaluating the model for the training and testing sets is shown.

Musical Attributes	Audio Features	Training AUC/ R^2	Testing AUC/ R^2	Label Type
Solo Brass	Timbre	0.796	0.798	binary
Piano	Timbre	0.721	0.716	binary
Reeds	Timbre	0.790	0.789	binary
Aux Percussion	Timbre	0.750	0.750	binary
FeelSwing	Rhythm	0.907	0.902	binary
FeelShuffle	Rhythm	0.919	0.920	binary
FeelSyncopation	Rhythm	0.772	0.770	binary
FeelBackBeat	Rhythm	0.400	0.393	continuous
FeelDance	Rhythm	0.527	0.515	continuous

Table 8. The results for learning binary (AUC) and continuous (R^2) attributes important to Jazz are shown.

First of all, we see that testing and training AUC is almost identical. Because of this, a single trial (fold) is appropriate when learning attribute models. The learned models should generalize over all music without over fitting. This justifies using the the same 70%:30% (train:test) split for each layer in the stacked models. We see that MFCC's do somewhat well for brass and reeds, but the lower AUC overall shows that these timbre features are not doing enough to capture these attributes, which may be a source of error in genre models that rely heavily on timbre. However, the rhythm results are much better, especially for swing and shuffle, which was argued in Section 4 and Table 5 as an important distinction to make when predicting Jazz subgenres.

Attribute Type	Num	Mean	Median	Maximum
Continuous Rhythm (R^2)	3	0.432 \pm 0.077	0.393	0.515
Continuous Timbre (R^2)	12	0.266 \pm 0.192	0.194	0.514
All Continuous	15	0.299 \pm 0.186	0.389	0.515
Binary Rhythm (AUC)	7	0.889 \pm 0.059	0.902	0.946
Binary Timbre (AUC)	26	0.794 \pm 0.074	0.794	0.925
All Binary	15	0.814 \pm 0.080	0.806	0.946

Table 9. Overall summary of learned attributes.

Table 9 shows a summary of learning the all of the selected 48 attributes from audio features. It shows similar trends to Table 8, with rhythmic attributes better described by audio features than timbral attributes. Furthermore, the continuous timbral attributes, which are sometimes complicated perceptually (e.g., vocal grittiness), are not modeled very well at all. This suggests that MFCC's, and possibly other spectral approximations, do not provide the full picture into what we perceive as the components of timbre. This is especially true in the context of instrument identification in mixtures, which is a main utility of the timbre features in this context. While these models as a whole can be improved, the problems of instrument identification and rhythm analysis are separate, large, and active research areas [14, 15, 25, 26].

6. CONCLUSION

In this work, we demonstrated that there is potential to demystify the constructs of musical genre into distinct musical components. The attributes we selected from music experts are able to provide a great deal of genre distinguishing information, but this is only an initial investigation into these questions. We were also able to discover and outline the importance of certain attributes in specific contexts. This strongly suggests that the expression of musical attributes are necessary additions to definitions of genre.

It was also shown here (and in previous work [25]) that audio features motivated by timbre and rhythm are, with some success, able to model musical attributes. Audio features are also able to describe musical genre directly and through stacked approaches that exploit the learned models of musical attributes. This is strong evidence suggesting that audio-based approaches are learning the presence of the musical attributes, to some degree, when distinguishing genre. In some cases, the audio-based models were more powerful than the human musical attribute models. This suggests that there is more to genre than our chosen subset of rhythm and orchestration attributes, and it makes us contemplate that there is more about the definition of genre yet to be discovered.

In seeking to improve on this work, we next look to investigate replacing the feature concatenation with late fusion of context-dependent classifiers (e.g., rhythm, timbre), which has shown improved results for genre classification [11]. It may also be helpful to use a greater number of the available attributes than the chosen 48, as well as additional attribute types (e.g., melody, harmony). Furthermore, perhaps the most interesting direction is to treat each musical attribute model as a hidden layer in a neural network. In these cases, the models that are trained to predict musical attributes will serve as a form of domain-specific pre-training. These models would perform full back propagation across an additional layer which connects our attributes to genres. This will potentially help to learn better models of genre as well as adjust the models of musical attributes in order better capture their genre relationships.

7. REFERENCES

- [1] Jean-Julien Aucouturier and François Pachet. Representing musical genre: A state of the art. *Journal of New Music Research*, 32(1):83–93, 2003.
- [2] Thierry Bertin-Mahieux, Douglas Eck, and Michael Mandel. Automatic tagging of audio: The state-of-the-art. *Machine audition: Principles, algorithms and systems*, pages 334–352, 2010.
- [3] Sebastian Böck and Gerhard Widmer. Maximum filter vibrato suppression for onset detection. In *Proc. of the 16th International Conference on Digital Audio Effects (DAFx-13)*, 2013.
- [4] Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.
- [5] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *Proc. of the 23rd international conference on Machine learning*, pages 233–240. ACM, 2006.
- [6] Li Deng and Dong Yu. Deep convex net: A scalable architecture for speech pattern classification. In *Proc. of Interspeech*, 2011.
- [7] Simon Dixon, Elias Pampalk, and Gerhard Widmer. Classification of dance music by periodicity patterns. In *Proc. of the International Society for Music Information Retrieval Conference*, 2003.
- [8] Douglas Eck, Paul Lamere, Thierry Bertin-Mahieux, and Stephen Green. Automatic generation of social tags for music recommendation. In *Advances in neural information processing systems*, pages 385–392, 2008.
- [9] Daniel PW Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, 2007.
- [10] Franco Fabbri. A theory of musical genres: Two applications. *Popular music perspectives*, 1:52–81, 1982.
- [11] Arthur Flexer, Fabien Gouyon, Simon Dixon, and Gerhard Widmer. Probabilistic combination of features for music classification. In *Proc. of the International Society for Music Information Retrieval Conference*, pages 111–114, 2006.
- [12] Fabien Gouyon and Simon Dixon. Dance music classification: A tempo-based approach. In *Proc. of the International Society for Music Information Retrieval Conference*, 2004.
- [13] Fabien Gouyon, Simon Dixon, Elias Pampalk, and Gerhard Widmer. Evaluating rhythmic descriptors for musical genre classification. In *Proc. of the AES 25th International Conference*, pages 196–204, 2004.
- [14] Philippe Hamel, Sean Wood, and Douglas Eck. Automatic identification of instrument classes in polyphonic and poly-instrument audio. In *Proc. of the International Society for Music Information Retrieval Conference*, pages 399–404, 2009.
- [15] Perfecto Herrera-Boyer, Geoffroy Peeters, and Shlomo Dubnov. Automatic classification of musical instrument sounds. *Journal of New Music Research*, 32(1):3–21, 2003.
- [16] André Holzapfel and Yannis Stylianou. Scale transform in rhythmic similarity of music. *IEEE Trans. on Audio, Speech and Language Processing*, 19(1):176–185, 2011.
- [17] S Jothilakshmi and N Kathiresan. Automatic music genre classification for indian music. In *Proc. Int. Conf. Software Computer App*, 2012.
- [18] Peter Knees, Tim Pohle, Markus Schedl, and Gerhard Widmer. Combining audio-based similarity with web-based data to accelerate automatic music playlist generation. In *Proc. of the 8th ACM international workshop on Multimedia information retrieval*, pages 147–154. ACM, 2006.
- [19] Miguel Lopes, Fabien Gouyon, Alessandro L Koerich, and Luiz ES Oliveira. Selection of training instances for music genre classification. In *Proc. of the International Conference on Pattern Recognition*, pages 4569–4572. IEEE, 2010.
- [20] F. Pachet and P. Roy. Improving multilabel analysis of music titles: A large-scale validation of the correction approach. *IEEE Trans. on Audio, Speech and Language Processing*, 17(2):335–343, 2009.
- [21] François Pachet and Daniel Cazaly. A taxonomy of musical genres. In *Content-Based Multimedia Information Access Conference*, pages 1238–1245, 2000.
- [22] Elias Pampalk, Arthur Flexer, and Gerhard Widmer. Improvements of audio-based music similarity and genre classification. In *Proc. of the International Society for Music Information Retrieval Conference*, volume 5, pages 634–637, 2005.
- [23] Maria Panteli, Niels Bogaards, and Aline Honingh. Modeling rhythm similarity for electronic dance music. *Proc. of the International Society for Music Information Retrieval Conference*, 2014.
- [24] Geoffroy Peeters. Rhythm classification using spectral rhythm patterns. In *Proc. of the International Society for Music Information Retrieval Conference*, pages 644–647, 2005.
- [25] Matthew Prockup, Andreas F. Ehmann, Fabien Gouyon, Erik M. Schmidt, and Youngmoo E. Kim. Modeling musical rhythm at scale using the music genome project. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2015.
- [26] Jeffrey Scott and Youngmoo E Kim. Instrument identification informed multi-track mixing. In *Proc. of the International Society for Music Information Retrieval Conference*, pages 305–310, 2013.
- [27] Klaus Seyerlehner, Markus Schedl, Peter Knees, and Reinhard Sonleitner. A refined block-level feature set for classification, similarity and tag prediction. *Extended Abstract to MIREX*, 2011.
- [28] Bob L Sturm. The state of the art ten years after a state of the art: Future research in music information retrieval. *Journal of New Music Research*, 43(2):147–172, 2014.
- [29] Derek Tingle, Youngmoo E Kim, and Douglas Turnbull. Exploring automatic music annotation with acoustically-objective tags. In *Proc. of the international conference on Multimedia information retrieval*, pages 55–62. ACM, 2010.
- [30] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Trans. on Audio, Speech and Language Processing*, 10(5):293–302, 2002.

COVER SONG IDENTIFICATION WITH TIMBRAL SHAPE SEQUENCES

Christopher J. Tralie

Duke University Department of
Electrical and Computer Engineering
chris.tralie@gmail.com

Paul Bendich

Duke University Department of
Mathematics
bendich@math.duke.edu

ABSTRACT

We introduce a novel low level feature for identifying cover songs which quantifies the relative changes in the smoothed frequency spectrum of a song. Our key insight is that a sliding window representation of a chunk of audio can be viewed as a time-ordered point cloud in high dimensions. For corresponding chunks of audio between different versions of the same song, these point clouds are approximately rotated, translated, and scaled copies of each other. If we treat MFCC embeddings as point clouds and cast the problem as a relative shape sequence, we are able to correctly identify 42/80 cover songs in the “Covers 80” dataset. By contrast, all other work to date on cover songs exclusively relies on matching note sequences from Chroma derived features.

1. INTRODUCTION

Automatic cover song identification is a surprisingly difficult classical problem that has long been of interest to the music information retrieval community [5]. This problem is significantly more challenging than traditional audio fingerprinting because a combination of tempo changes, musical key transpositions, embellishments in time and expression, and changes in vocals and instrumentation can all occur simultaneously between the original version of a song and its cover. Hence, low level features used in this task need to be robust to all of these phenomena, ruling out raw forms of popular features such as MFCC, CQT, and Chroma.

One prior approach, as reviewed in Section 2, is to compare beat-unsynchronous sequences of chroma vectors between candidate covers. The beat-syncing helps this be invariant to tempo, but it is still not invariant to key. However, many schemes have been proposed to deal with this, up to and including a brute force check over all key transpositions.

Chroma representations factor out some timbral information by folding together all octaves, which is sensible given the effect that different instruments and recording environments have on timbre. However, valuable non-pitch

information which is preserved between cover versions, such as spectral fingerprints from drum patterns, is obscured in Chroma representation. This motivated us to take another look at whether timbral-based features could be used at all for this problem. Our idea is that even if absolute timbral information is vastly different between two versions of the same song, the *relative evolution* of timbre over time should be comparable.

With careful centering and normalization within small windows to combat differences in global timbral drift between the two songs, we are indeed able to design shape features which are approximately invariant to cover. These features, which are based on self-similarity matrices of MFCC coefficients, can be used on their own to effectively score cover songs. This, in turn, demonstrates that even if absolute pitch is obscured and blurred, cover song identification is still possible.

Section 2 reviews prior work in cover song identification. Our method is described in detail by Sections 3 and 4. Finally, we report results on the “Covers 80” benchmark dataset [7] in Section 5, and we apply our algorithm to the recent “Blurred Lines” copyright controversy.

2. PRIOR WORK

To the best of our knowledge, all prior low level feature design for cover song identification has focused on Chroma-based representations alone. The cover songs problem statement began with the work of [5], which used FFT-based cross-correlation of all key transpositions of beat-synchronous chroma between two songs. A follow-up work [8] showed that high passing such cross-correlation can lead to better results. In general, however, cross-correlation is not robust to changes in timing, and it is also a global alignment technique. Serra [22] extended this initial work by considering dynamic programming local alignment of chroma sequences, with follow-up work and rigorous parameter testing and an “optimal key transposition index” estimation presented in [23]. The same authors also showed that a delay embedding of statistics spanning multiple beats before local alignment improves classification accuracy [25]. In a different approach, [14] compared modeled covariance statistics of all chroma bins, as well as comparing covariance statistics for all pairwise differences of beat-level chroma features, which is not unlike the “bag of words” and bigram representations, respectively, in text analysis. Other work tried to model sequences of



© Christopher J. Tralie, Paul Bendich.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Christopher J. Tralie, Paul Bendich. “Cover Song Identification with Timbral Shape Sequences”, 16th International Society for Music Information Retrieval Conference, 2015.

chords [2] as a slightly higher level feature than chroma. Slightly later work concentrated on fusing the results of music separated into melody and accompaniment [11] and melody, bass line, and harmony [21], showing improvements over matching chroma on the raw audio. The most recent work on cover song identification has focused on fast techniques for large scale pitch-based cover song identification, using a sparse set of approximate nearest neighbors [28] and low dimensional projections [12]. Authors in [9] and [17] also use the *magnitude* of the 2D Fourier Transform of a sequences of chroma vectors treated as an image, so the resulting coefficients will be automatically invariant to key and time shifting without any extra computation, at the cost of some discriminative power.

Outside of cover song identification, there are other works which examine gappy sequences of MFCC in music, such as [4]. However, these works look at matched sequences of MFCC-like features in their original feature space. By contrast, in our work, we examine the *relative* shape of such features. Finally, we are not the first to consider shape in an applied musical context. For instance, [29] turns sequences of notes in sheet music into plane curves, whose curvature is then examined. To our knowledge, however, we are the first to explicitly model shape in musical audio for version identification.

3. TIME ORDERED POINT CLOUDS FROM BLOCKS OF AUDIO

The first step of our algorithm uses a timbre-based method to turn a block of audio into what we call a *time-ordered point cloud*. We can then compare to other time-ordered point clouds in a rotation, translation, and scale invariant manner using normalized Euclidean Self-Similarity matrices (Section 3.3). The goal is to then match up the relative shape of musical trajectories between cover versions.

3.1 Point Clouds from Blocks and Windows

We start with a song, which is a function of time $f(t)$ that has been discretized as some vector X . In the following discussion, the symbol $X(a, b)$ means the song portion beginning at time $t = a$ and ending at time $t = b$. Given X , there are many ways to summarize a chunk of audio $w \in X$, which we call a *window*, as a point in some feature space. We use the classical Mel-Frequency Cepstral coefficient representation [3], which is based on a perceptually motivated log frequency and log power short-time Fourier transform that preserves timbral information. In our application, we perform an MFCC with 20 coefficients, giving rise to a 20-dimensional point.

$$MFCC(w) \in \mathbb{R}^{20} \quad (1)$$

Given a longer chunk of audio, which we call a *block*, we can use the above embedding on a collection of K windows that cover the block to construct a collection of points, or a *point cloud*, representing that block. More formally, given a block covering a range $[t_1, t_2]$, we want a set of window intervals $[a_i, b_i]$, with $i = 1..K$, so that

- $a_i < b_i$
- $a_i < a_{i+1}, b_i < b_{i+1}$
- $\cup_{i=1}^K [a_i, b_i] = [t_1, t_2]$

Where t_1, t_2, a_i , and b_i are all discrete time indices into the sampled audio X . Hence, our final operator takes a set of time-ordered intervals $\{[a_1, b_1], [a_2, b_2], \dots, [a_K, b_K]\}$ which cover a block $[t_1, t_2]$ and turns them into a K -dimensional point cloud in \mathbb{R}^{20}

$$PC(\{[a_1, b_1], \dots, [a_K, b_K]\}) = \{MFCC(X(a_1, b_1)), \dots, MFCC(X(a_K, b_K))\} \quad (2)$$

3.2 Beat-Synchronous Blocks

As many others in the MIR community have done, including [5] and [8] for the cover songs application, we compute our features synchronized within beat intervals. We use a simple dynamic programming beat tracker developed in [6]. Similarly to [8], we bias the beat tracker with three initial tempo levels: 60BPM, 120BPM, and 180BPM, and we compare the embeddings from all three levels against each other when comparing two songs, taking the best score out of the 9 combinations. This is to mitigate the tendency of the beat tracker to double or halve the true beat intervals of different versions of the same song when there are tempo changes between the two. The trade-off is of course additional computation. We should note that other cover song works, such as [23], avoid beat tracking step altogether, hence bypassing these problems. However, it is important for us to align our sequences as well as possible in time so that shape features are in correspondence, and this is a straightforward way to do so.

Given a set of beat intervals, the union of which makes up the entire song, we take blocks to be all contiguous groups of B beat intervals. In other words, we create a sequence of overlapping blocks X_1, X_2, \dots such that X_i is made up of B time-contiguous beat intervals, and X_i and X_{i+1} differ only by the starting beat of X_i and the finishing beat of X_{i+1} . Hence, given N beat intervals, there are $N - B + 1$ blocks total. Note that computing an embedding over more than one beat is similar in spirit to the chroma delay embedding approach in [25]. Intuitively, examining patterns over a group of beats gives more information than one beat alone, the effect of which is empirically evaluated in Section 5. For all blocks, we take the window size W to be the length of the average tempo period, and we advance the window intervals evenly from the beginning of the block to the end of a block with a *hop size* $H = W/200$. Hence, there is a 99.5% overlap between windows. We were inspired by theory on raw 1D time series signals [18], which shows that matching the window length to be just under the length of the period in a delay embedding maximizes the roundness of the embedding. Here we would like to match beat-level periodicities and fluctuations therein, so it is sensible to choose a window size corresponding to the tempo. This is in contrast to most other applications that use MFCC sliding window embeddings, which use a much smaller window size on the

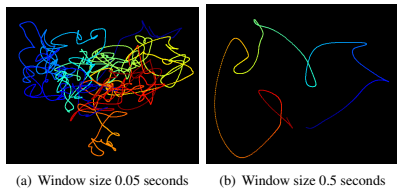


Figure 1. A screenshot from our GUI showing PCA on the sliding window representation of an 8-beat block from the hook of Robert Palmer’s “Addicted To Love” with two different window sizes. Cool colors indicate windows towards the beginning of the block, and hot colors indicate windows towards the end.

order of 10s of milliseconds, generally with a 50% overlap, to ensure that the frequency statistics are stationary in each window. In our application, however, we have found that a longer window size makes our self similarity matrices (Section 3.3) smoother, allowing for more reliable matches of beat-level musical trajectories, while having more windows per beat (high overlap) leads to more robust matching of SSMs using L2 (Section 4.1).

Figure 1 shows the first three principal components of an MFCC embedding with a traditional small window size versus our longer window embedding to show the smoothing effect.

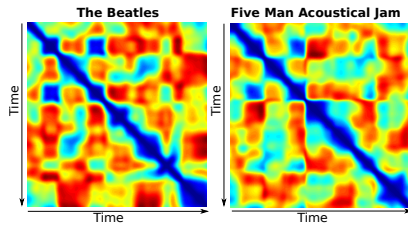
3.3 Euclidean Self-Similarity Matrices

For each beat-synchronous block X_t spanning B beats, we have a 20-dimensional point cloud extracted from the sliding window MFCC representation. Given such a time-ordered point cloud, there is a natural way to create an image which represents the shape of this point cloud in a rotation and translation invariant way, called the *self-similarity matrix* (SSM) representation.

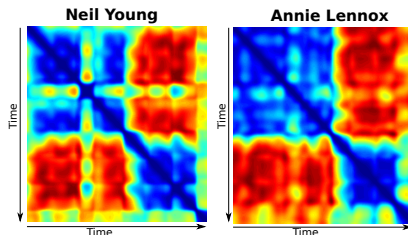
Definition 1. A *Euclidean Self-Similarity Matrix* (SSM) over an ordered point cloud $X_t \in \mathbb{R}^{M \times k}$ is an $M \times M$ matrix D so that

$$D_{ij} = \|X_t[i] - X_t[j]\|_2 \quad (3)$$

In other words, an SSM is an image representing all pairwise distances between points in a point cloud ordered by time. SSMs have been used extensively in the MIR community already, spearheaded by the work of Foote in 2000 for note segmentation in time [10]. They are now often used in general segmentation tasks [24] [15]. They have also been successfully applied in other communities, such as computer vision to recognize activity classes in videos from different points of view and by different actors [13]. Inspired by this work, we use self-similarity matrices as isometry invariant descriptors of local shape in our sliding windows of beat blocks, with the goal of capturing relative shape. In our case, the “activities” are musical expressions over small intervals, and the “actors” are different performers or groups of instruments.



(a) A block of 4 beats with 400 windows sliding in the song “We Can Work It Out” by The Beatles with a cover by Five Man Acoustical Jam



(b) A block of 4 beats with 400 windows sliding in the song “Don’t Let It Bring You Down” by Neil Young with a cover by Annie Lennox.

Figure 2. Two examples of MFCC SSM blocks which were matched between a song and its cover in the covers80 dataset. Hot colors indicate windows in the block are far from each other, and cool colors indicate that they are close.

To help normalize for loudness and other changes in relationships between instruments, we first center the point cloud within each block on its mean and scale each point to have unit norm before computing the SSM. That is, we compute the SSM on \hat{X}_t , where

$$\hat{X}_t = \left\{ \frac{x - \text{mean}(x)}{\|x - \text{mean}(x)\|_2} : x \in X_t \right\} \quad (4)$$

Also, not every beat block has the same number of samples due to natural variations of tempo in real songs. Thus, to allow comparisons between all blocks, we resize each SSM to a common image dimension $d \times d$, which is a parameter chosen in advance, the effects of which are explored empirically in Section 5.

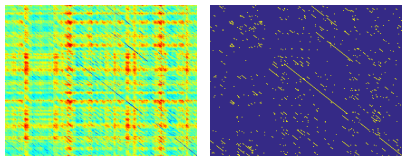
Figure 2 shows examples of SSMs of 4-beat blocks pulled from the Covers80 dataset that our algorithm matches between two different versions of the same song. Visually, similarities in the matched regions are evident. In particular, viewing the images as height functions, many of the critical points are close to each other. The “We Can Work It Out” example shows how this can work even for live performances, where the overall acoustics are quite different. Even more strikingly, the “Don’t Let It Bring You Down” example shows how similar shape patterns emerge even with an opposite gender singer and radically different instrumentation. Of course, in both examples,

there are subtle differences due to embellishments, local time stretching, and imperfect normalization between the different versions, but as we show in Section 5, there are often enough similarities to match up blocks correctly in practice.

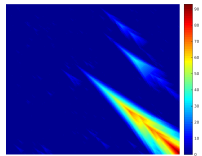
4. GLOBAL COMPARISON OF TWO SONGS

Once all of the beat-synchronous SSMS have been extracted from two songs, we do a global comparison between all SSMS from two songs to score them as cover matches. Figure 3 shows a block diagram of our system. After extracting beat-synchronous timbral shape features on SSMS, we then extract a binary cross-similarity matrix based on the L2 distance between all pairs of self-similarity matrices between two songs. We subsequently apply the Smith Waterman algorithm on the binary cross-similarity matrix to score a match between the two songs.

4.1 Binary Cross-Similarity And Local Alignment



(a) Full cross-similarity matrix (CSM) (b) 212×212 Binary cross-similarity matrix (B^M) with $\kappa = 0.05$



(c) Smith Waterman with local constraints: Score 93.1

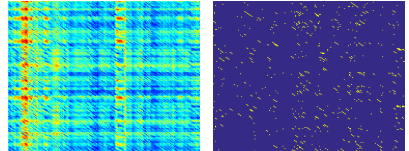
Figure 4. Cross-similarity matrix and Smith Waterman on MFCC-based SSMS for a true cover song pair of “We Can Work It Out” by The Beatles and Five Man Acoustical Jam.

Given a set of N beat-synchronous block SSMS for a song A and a set of M beat-synchronous block SSMS for a song B, we compute a song-level matching between song A and B by comparing all pairs of SSMS between the two songs. For this we create an $N \times M$ cross-similarity matrix (CSM), where

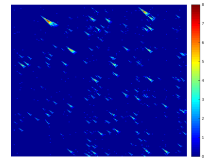
$$CSM_{ij} = \|\text{SSMA}_i - \text{SSMB}_j\|_2 \quad (5)$$

is the Frobenius norm (L2 image norm) between the SSM for the i^{th} beat block from song A and the SSM for j^{th} beat block for song B. Given this cross-similarity information, we then compute a binary cross similarity matrix B^M . A binary matrix is necessary so that we can apply the Smith Waterman local alignment algorithm [27] to score the match between song A and B, since Smith Waterman

only works on a discrete, quantized alphabet, not real values [23]. To compute B^M , we take the mutual fraction κ nearest neighbors between song A and song B, as in [25]. That is, $B_{ij}^M = 1$ if CSM_{ij} is within the κM^{th} smallest values in row i of the CSM and if CSM_{ij} is within the κN^{th} smallest values in column j of the CSM, and 0 otherwise. As in [25], we found that a dynamic distance threshold for mutual nearest neighbors per element worked significantly better than a fixed distance threshold for the entire matrix.



(a) Full cross-similarity matrix (CSM) (b) 212×185 Binary cross-similarity matrix (B^M) with $\kappa = 0.05$



(c) Smith Waterman with local constraints: Score 8

Figure 5. Cross-similarity matrix and Smith Waterman on MFCC-based SSMS for two songs that are not covers of each other: “We Can Work It Out” by The Beatles and “Yesterday” by En Vogue.

Once we have the B^M matrix, we can feed it to the Smith Waterman algorithm, which finds the best local alignment between the two songs, allowing for time shifting and gaps. Local alignment is a more appropriate choice than global alignment for the cover songs problem, since it is possible that different versions of the same song may have intros, outros, or bridge sections that were not present in the original song, but otherwise there are many sections in common. We choose a version of Smith Waterman with diagonal constraints, which was shown to work well for aligning binary cross-similarity matrices for chroma in cover song identification [23]. In particular, we recursively compute a matrix D so that

$$D_{ij} = \max \left\{ \begin{array}{l} D_{i-1,j-1} + (2\delta(B_{i-1,j-1}) - 1) + \Delta(B_{i-2,j-2}, B_{i-1,j-1}), \\ D_{i-2,j-1} + (2\delta(B_{i-1,j-1}) - 1) + \Delta(B_{i-3,j-2}, B_{i-1,j-1}), \\ D_{i-1,j-2} + (2\delta(B_{i-1,j-1}) - 1) + \Delta(B_{i-2,j-3}, B_{i-1,j-1}), \\ 0 \end{array} \right. \quad (6)$$

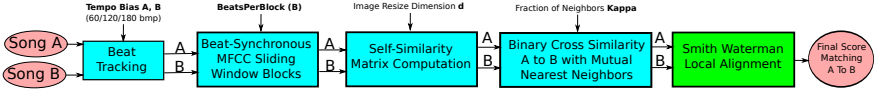


Figure 3. A block diagram of our system for computing a cover song similarity score of two songs using timbral features.

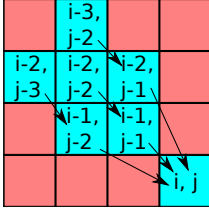


Figure 6. Constrained local matching paths considered in Smith Waterman, as prescribed by [23].

where δ is the Kronecker delta function and

$$\Delta(a, b) = \begin{cases} 0 & b = 1 \\ -0.5 & b = 0, a = 1 \\ -0.7 & b = 0, a = 0 \end{cases} \quad (7)$$

The $(2\delta(B_{i-1,j-1}) - 1)$ term in each line is such that there will be a +1 score for a match and a -1 score for a mismatch. The Δ function is the so-called “affine gap penalty” which gives a score of $-0.5 - 0.7(g - 1)$ for a gap of length g . The local constraints are to bias Smith Waterman to choosing paths along near-diagonals of B^M . This is important since in musical applications, we do not expect large gaps in time in one song that are not in the other, which would show up as horizontal or vertical paths through the B^M matrix. Rather, we prefer gaps that occur nearly simultaneously in time for a poorly matched beat or set of beats in an otherwise well-matching section. Figure 6 shows a visual representation of the paths considered through B^M .

Figure 4 shows an example of a CSM, B^M , and resulting Smith Waterman for a true cover song pair. Several long diagonals are visible, indicating large chunks of the two songs are in correspondence, and this gives rise to a large score of 93.1 between the two songs. Figure 5 shows the CSM, B , and Smith Waterman for two songs which are not versions of each other. By contrast, there are no long diagonals, and this pair only receives a score of 8.

5. RESULTS

To benchmark our algorithm, we apply it to the standard “Covers 80” dataset [7], which consists of 80 sets of two versions of the same song, most of which are pop songs from the past three decades. There are designated two sets of songs A and B, each with exactly one version of every pair. To benchmark our algorithm on this dataset, we follow the scheme in [5] and [8]. That is, given a song from set A, compute the Smith Waterman score from all songs

from set B and declare the cover song to be the one with the maximum score. Note that a random classifier would only get 1/80 in this scheme. The best scores reported on this dataset are 72/80 [20], using a support vector machine on several different chroma-derived features.

Table 1 shows the correctly identified songs based on the maximum score, given variations of the parameters we have in our algorithm. We achieve a maximum score of 42/80 for a variety of parameter combinations. The nearest neighbor fraction κ and the dimension of the SSM image have very little effect, but increasing the number of beats per block has a positive effect on the performance. The stability of κ and d are encouraging from a robustness standpoint, and the positive effect increasing the number of beats per block suggests that the shape of medium scale musical expressions are more discriminative than smaller ones.

Table 1. The number of songs that are correctly ranked as the most similar in the Covers 80 dataset, varying parameters. κ is the nearest neighbor fraction, B is the number of beats per block, and d is the resized dimension of the Euclidean Self-Similarity images.

Kappa = 0.05	B = 8	B = 10	B = 12	B = 14
d = 100	30	33	36	40
d = 200	31	33	36	39
d = 300	31	34	36	40
Kappa = 0.1	B = 8	B = 10	B = 12	B = 14
d = 100	35	39	41	42
d = 200	36	38	42	42
d = 300	36	38	41	41
Kappa = 0.15	B = 8	B = 10	B = 12	B = 14
d = 100	36	42	41	42
d = 200	36	41	41	42
d = 300	38	42	42	41

In addition to the Covers 80 benchmark, we apply our cover songs score to a recent popular music controversy, the “Blurred Lines” controversy [16]. Marvin Gaye’s estate argues that Robin Thicke’s recent pop song “Blurred Lines” is a copyright infringement of Gaye’s “Got To Give It Up.” Though the note sequences differ between the two songs, ruling out any chance of a high chroma-based score, Robin Thicke has said that his song was meant to “evoke an era” (Marvin Gaye’s era) and that he derived significant inspiration from “Got To Give It Up” specifically [16]. Without making a statement about any legal implications, we note that our timbral shape-based score between “Blurred Lines” and “Got To Give It Up” is in the 99.9th percentile of all scores between songs in group A and group B in the

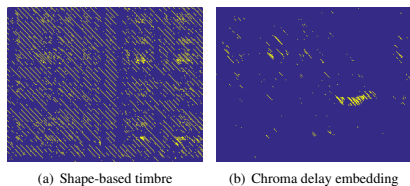


Figure 7. Corresponding portions of the binary cross-similarity matrix between Marvin Gaye’s “Got To Give It Up” and Robin Thicke’s “Blurred Lines” for both shape-based timbre (our technique) and chroma delay embedding

Covers 80 dataset, for $\kappa = 0.1$, $B = 14$, and $d = 200$. Unsurprisingly, when comparing “Blurred Lines” with all other songs in the Covers 80 database plus “Got To Give It Up,” “Got To Give It Up” was the highest ranked. For reference, binary cross similarity matrices are shown in Figure 7, both for our timbre shape based technique and the delay embedding chroma technique in [25]. The timbre-based cross-similarity matrix is densely populated with diagonals, while the pitch-based one is not.

6. CONCLUSIONS AND FUTURE WORK

We show that timbral information in the form of MFCC can indeed be used for cover song identification. Most prior approaches have used Chroma-based features averaged over intervals. By contrast, we show that an analysis of the fine relative shape of MFCC features over intervals is another way to achieve good performance. This opens up the possibility for MFCC to be used in much more flexible music information retrieval scenarios than traditional audio fingerprinting.

On the more technical side, we should note that for comparing shape, L2 of SSMs for cross-similarity is fairly simple and not robust to local re-parameterizations in time between versions, though we tried many other isometry invariant shape descriptors that were significantly slower and yielded inferior performance in initial implementation. In particular, we tried curvature descriptors (ratio of arc length to chord length), Gromov-Hausdorff distance after fractional iterative closest points aligning MFCC block curves [19], and Earth Mover’s distance between SSMs [26]. If we are able to find another shape descriptor which performs better than our current scheme but is slower, we may still be able to make it computationally feasible by using the “Generalized Patch Match” algorithm [1] to reduce the number of pairwise block comparisons needed by exploiting coherence in time. This is similar in spirit to the approximate nearest neighbors schemes proposed in [28] for large scale cover song identification, and we could adapt their sparse Smith Waterman algorithm to our problem. In an initial implementation of generalized patch match for our current scheme, we found we only needed to query about 15% of the block pairs.

7. SUPPLEMENTARY MATERIAL

We have documented our code and uploaded directions for performing all experiments run in this paper. We also created an open source graphical user interface which can be used to interactively view cross-similarity matrices and to examine the shape of blocks of audio after 3D PCA using OpenGL. All code can be found in the ISMIR2015 directory at

github.com/ctralie/PublicationsCode.

8. ACKNOWLEDGEMENTS

Chris Tralie was supported under NSF-DMS 1045133 and an NSF Graduate Fellowship. Paul Bendich was supported by NSF 144749. John Harer and Guillermo Sapiro are thanked for valuable feedback. The authors would also like to thank the Information Initiative at Duke (iiD) for stimulating this collaboration.

9. REFERENCES

- [1] Connelly Barnes, Eli Shechtman, Dan B Goldman, and Adam Finkelstein. The generalized patchmatch correspondence algorithm. In *Computer Vision—ECCV 2010*, pages 29–43. Springer, 2010.
- [2] Juan Pablo Bello. Audio-based cover song retrieval using approximate chord sequences: Testing shifts, gaps, swaps and beats. In *ISMIR*, volume 7, pages 239–244, 2007.
- [3] Bruce P Bogert, Michael JR Healy, and John W Tukey. The quefrency analysis of time series for echoes: Cepstrum, pseudo-autocovariance, cross-cepstrum and saphe cracking. In *Proceedings of the symposium on time series analysis*, volume 15, pages 209–243. chapter, 1963.
- [4] Michael Casey and Malcolm Slaney. The importance of sequences in musical similarity. In *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, volume 5, pages V–V. IEEE, 2006.
- [5] Daniel PW Ellis. Identifying ‘cover songs’ with beat-synchronous chroma features. *MIREX 2006*, pages 1–4, 2006.
- [6] Daniel PW Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, 2007.
- [7] Daniel PW Ellis. The “covers80” cover song data set. URL: <http://labrosa.ee.columbia.edu/projects/coversongs/covers80>, 2007.
- [8] Daniel PW Ellis and Courtenay Valentine Cotton. The 2007 labrosa cover song detection system. *MIREX 2007*, 2007.

- [9] Daniel PW Ellis and Bertin-Mahieux Thierry. Large-scale cover song recognition using the 2d fourier transform magnitude. In *The 13th international society for music information retrieval conference*, pages 241–246, 2012.
- [10] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. In *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, volume 1, pages 452–455. IEEE, 2000.
- [11] Rémi Foucard, J-L Durrieu, Mathieu Lagrange, and Gaël Richard. Multimodal similarity between musical streams for cover version detection. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 5514–5517. IEEE, 2010.
- [12] Eric J Humphrey, Oriol Nieto, and Juan Pablo Bello. Data driven and discriminative projections for large-scale cover song identification. In *ISMIR*, pages 149–154, 2013.
- [13] Imran N Junejo, Emilie Dexter, Ivan Laptev, and Patrick Pérez. Cross-view action recognition from temporal self-similarities. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, pages 293–306. Springer-Verlag, 2008.
- [14] Samuel Kim, Erdem Unal, and Shrikanth Narayanan. Music fingerprint extraction for classical music cover song identification. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 1261–1264. IEEE, 2008.
- [15] Brian McFee and Daniel PW Ellis. Analyzing song structure with spectral clustering. In *15th International Society for Music Information Retrieval (ISMIR) Conference*, 2014.
- [16] Emily Miao and Nicole E Grimm. The blurred lines of what constitutes copyright infringement of music: Robin thicke v. marvin gayes estate. *WESTLAW J. INTELLECTUAL PROP.*, 20:1, 2013.
- [17] Oriol Nieto and Juan Pablo Bello. Music segment similarity using 2d-fourier magnitude coefficients. In *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, pages 664–668. IEEE, 2014.
- [18] Jose A Perea and John Harer. Sliding windows and persistence: An application of topological methods to signal analysis. *Foundations of Computational Mathematics*, pages 1–40, 2013.
- [19] Jeff M Phillips, Ran Liu, and Carlo Tomasi. Outlier robust icp for minimizing fractional rmsd. In *3-D Digital Imaging and Modeling, 2007. 3DIM'07. Sixth International Conference on*, pages 427–434. IEEE, 2007.
- [20] Suman Ravuri and Daniel PW Ellis. Cover song detection: from high scores to general classification. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 65–68. IEEE, 2010.
- [21] Justin Salamon, Joan Serrà, and Emilia Gómez. Melody, bass line, and harmony representations for music version identification. In *Proceedings of the 21st international conference companion on World Wide Web*, pages 887–894. ACM, 2012.
- [22] J Serra. Music similarity based on sequences of descriptors: tonal features applied to audio cover song identification. *Department of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain*, 2007.
- [23] Joan Serra, Emilia Gómez, Perfecto Herrera, and Xavier Serra. Chroma binary similarity and local alignment applied to cover song identification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(6):1138–1151, 2008.
- [24] Joan Serra, Meinard Müller, Peter Grosche, and Josep Lluís Arcos. Unsupervised detection of music boundaries by time series structure features. In *Twenty-Sixth AAAI Conference on Artificial Intelligence*, 2012.
- [25] Joan Serra, Xavier Serra, and Ralph G Andrzejak. Cross recurrence quantification for cover song identification. *New Journal of Physics*, 11(9):093017, 2009.
- [26] Sameer Shirdhonkar and David W Jacobs. Approximate earth movers distance in linear time. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.
- [27] Temple F Smith and Michael S Waterman. Identification of common molecular subsequences. *Journal of molecular biology*, 147(1):195–197, 1981.
- [28] Romain Tavenard, Hervé Jégou, and Mathieu Lagrange. Efficient cover song identification using approximate nearest neighbors. 2012.
- [29] Julián Urbano, Juan Lloréns, Jorge Morato, and Sonia Sánchez-Cuadrado. Melodic similarity through shape similarity. In *Exploring music contents*, pages 338–355. Springer, 2011.

ON THE IMPACT OF KEY DETECTION PERFORMANCE FOR IDENTIFYING CLASSICAL MUSIC STYLES

Christof Weiß

Fraunhofer Institute for Digital Media Technology Ilmenau
christof.weiss@idmt.fraunhofer.de

Maximilian Schaab

Fraunhofer Institute for Digital Media Technology Ilmenau

ABSTRACT

We study the automatic identification of Western classical music styles by directly using chroma histograms as classification features. Thereby, we evaluate the benefits of knowing a piece’s global key for estimating key-related pitch classes. First, we present four automatic key detection systems. We compare their performance on suitable datasets of classical music and optimize the algorithms’ free parameters. Using a second dataset, we evaluate automatic classification into the four style periods Baroque, Classical, Romantic, and Modern. To that end, we calculate global chroma statistics of each audio track. We then split up the tracks according to major and minor keys and circularly shift the chroma histograms with respect to the tonic note. Based on these features, we train two individual classifier models for major and minor keys. We test the efficiency of four chroma extraction algorithms for classification. Furthermore, we evaluate the impact of key detection performance on the classification results. Additionally, we compare the key-related chroma features to other chroma-based features. We obtain improved performance when using an efficient key detection method for shifting the chroma histograms.

1. INTRODUCTION

In the field of Music Information Retrieval (MIR), a considerable amount of research has been performed to classify music audio recordings according to different categories [3, 29]. Beyond top-level *genres* such as Rock, Jazz, or Classical, several attempts towards resolving *subgenres* have been made. We dedicate ourselves to the subgenre classification of Western classical music which has been addressed sparsely in previous work.

There are plenty of possibilities to organize classical music archives. Apart from the specific artists—soloists or ensembles—, timbral properties such as the predominant instrument(s) may serve as categories [26]. We think that the rather abstract concept of *musical style* provides a

more appropriate subgenre taxonomy. The specific application of this idea leads to the task of composer identification [4, 11, 15, 22]. Beyond such a detailed taxonomy, we restrict ourselves to more general categories—the *historical periods* Baroque, Classical, Romantic, and Modern.¹ This naturally constitutes a simplification but may provide a convenient starting point for finer analyses [6].

Several researchers have published studies on the basis of symbolic data such as score or MIDI representations [1, 8, 10, 11, 15, 22, 25]. However, we find some benefits when directly dealing with audio recordings. First, the audio incorporates more information than the score by representing the “sounding reality” of the music to a higher degree.² Second, audio-based methods enable nice applications for organizing and browsing today’s large archives of classical music. Moreover, such archives provide precious possibilities for data-driven musicological research in a new quantitative dimension.

Studies based on symbolic data often make use of musical properties such as the use of specific intervals [1] or chords [22]. Sometimes, characteristics of polyphony and voice leading are considered as well [1, 11]. Other methods rely on more fundamental properties of harmony such as the occurrence of pitch classes [10] and pitch class sets [8]. Usually, researchers statistically analyze these characteristics to obtain classification features. These features are then used as input for machine learning (ML) classifiers.

There are several limitations for harmonic analysis of audio based on state-of-the-art signal processing algorithms. Due to the restricted performance of automatic music transcription³, we build our method upon chroma features that have been shown to suitably represent the pitch class content of audio [7, 19]. Using chroma features, several musical characteristics such as voice leading properties or interval and chord inversions cannot be resolved. Furthermore, acoustic phenomena such as overtones and timbre show considerable effect on the chroma features. Scholars proposed several attempts to approach these problems by enhancing the robustness of chroma [7, 13, 14, 17].

Researches have proposed several chroma-based fea-



© Christof Weiß, Maximilian Schaab.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Christof Weiß, Maximilian Schaab. “On the Impact of Key Detection Performance for Identifying Classical Music Styles”, 16th International Society for Music Information Retrieval Conference, 2015.

¹ Here, the *Modern* class refers to 20th century art music with some stylistic distance from romantic music.

² This observation particularly matters for older music such as the Baroque style, where numerous conventions for practical performance were known by the interpreters without notating them in the scores.

³ In particular, these algorithms highly depend on the orchestration. On that account, automatic transcription is not reliable when dealing with mixed music for piano, orchestra, and voices.

ture types for classifying musical genres and styles. Tzanetakis uses the predominant pitch class, its relative amplitude, and the size of the predominant interval as features [29]. Others extract chords from audio and classify based on the chord types and progressions [24]. In [32], interval and chord types are estimated from different chroma resolutions. Furthermore, measures for quantifying tonal complexity have been tested as classification features [33].

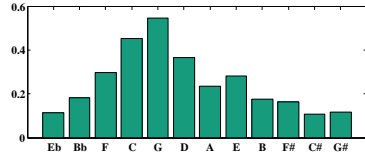
In this paper, we want to test a simpler approach that directly uses *chroma histograms* as input for a classifier (Section 2). For tonal music, the chroma distribution of a piece is mainly influenced by the *musical key*. Usually, the notes of the underlying scale and the most prominent chords obtain high values—such as the tonic note or the dominant note. We therefore test the benefits of knowing the global key for classifying with chroma features. To that end, we first compare four key detection methods (Section 2.2) on suitable datasets of classical music and optimize the algorithms’ parameters (Section 3.2). Second, we perform classification experiments on a separate dataset of 1600 classical pieces (Section 3.3). As classification features, we use chroma histograms that are shifted on the basis of different key algorithms or ground truth key annotations. We test the influence of considering the key as well as the effect of training separate models for major and minor keys. Finally, we compare these features’ performance against other chroma-based features introduced in earlier work [32,33].

2. PROPOSED METHOD

In Western classical music, tonality and harmony play a central part for establishing musical form, expression, and style. The use of specific pitches, intervals, and chords—as well as their progressions—constitute typical style markers. They hierarchically depend on each other and contribute to the chroma distribution of a piece. Beyond the high importance of the global key, modulations to other keys entail the use of different chords and pitches. That way, sections in foreign keys considerably contribute to the global chroma histogram—depending on their length. Apart from such harmonic characteristics, instrumentation and timbre may affect the shape of the chroma distribution. Let us consider a simple major triad: Depending on the instrumentation, the root, third, or fifth note may be more pronounced leading to different chroma vectors.

Some of these differences may serve to resolve subtler stylistic differences. In Figure 1, we show two chroma histograms of symphony movements by Schumann and Brahms, both in a major key and centered to their respective tonic note (C and F). Though these composers have much in common—a part of their lifetime, the cultural background, and several inspiring persons—the pieces considerably differ in their pitch class histograms. One reason may be the more complex harmony in Brahms’ music—the chromatic pitch classes such as F#, C#, and Ab are enhanced compared to Schumann’s equivalents. Moreover, Brahms’ instrumentation often emphasizes the chords’ third notes. This could explain the increased val-

Schumann, 2nd symphony, 1st mvmt. (C major)



Brahms, 3rd symphony, 1st mvmt. (F major)

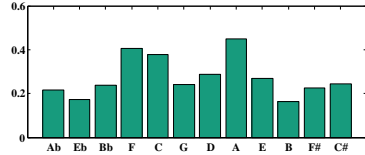


Figure 1. Chroma histograms for Schumann’s 2nd symphony, 1st movement in C major (upper plot) and Brahms 3rd symphony, 1st movement in F major (lower plot). The histograms are arranged according to the circle of fifths and centered to the respective tonic note. We normalize the distributions to the ℓ_2 norm in order to ensure comparability.

ues for D, A, and E—the triad thirds of the main chords BbM (subdominant), FM (tonic), and CM (dominant), respectively. Another explanation for this observation could be a modulation to the local key A major for a considerable amount of time. Such modulations to third-related *mediant keys* are common in late romantic music.

To describe such characteristics, the *relative pitch classes* are important. Therefore, we need information about the global key. Sometimes, this metadata is provided in musical archives. However, such annotations are often incomplete. For work cycles and multi-movement works, we usually find only one key (“Symphony in F major”) which single movements may differ from. For those reasons, we test automatic methods for audio key detection and evaluate the influence of their performance on the overall classification results. We also compare automatic key detection to the use of ground truth key annotations.

Apart from the tonic note, the mode (major / minor) is of high importance, since the harmonic structure of minor pieces fundamentally differs from the one in major. To that end, we split up our data and train a separate model for each mode. Section 2.3 outlines the details of this idea.

2.1 Chroma Features

In audio signal processing, chroma features have been shown to suitably represent tonal characteristics [7, 19]. For a chromagram, the spectrogram bins are mapped into a series of 12-dimensional chroma vectors $\mathbf{c} = (c_0, c_1, \dots, c_{11})^T \in \mathbb{R}^{12}$. These vectors represent the energy of the pitch classes that are independent from the octave. To reduce the influence of overtones and timbral characteristics, several chroma extraction methods have been proposed. We consider six different approaches:

- (i) **CP.** This algorithm [21] is based on a multirate filter bank and published in the Chroma Toolbox [18]. We use the *Chroma Pitch* as our baseline feature.
- (ii) **CLP.** Jiang et al. found improvement of chord recognition when using logarithmic compression before octave summarization. We use the *Chroma Logarithmic Pitch* with compression parameter $\eta = 1000$ which performed best in [9].
- (iii) **CRP.** Müller and Ewert proposed a method to eliminate timbral information using the Discrete Cosine Transform—*Chroma-DCT-Reduced Log Pitch* [17].
- (iv) **HPCP.** These *Harmonic Pitch Class Profiles* consider the overtones for the chroma computation [7].
- (v) **EPCP.** In [27], *Enhanced Pitch Class Profiles* [13] performed best in a chord matching experiment. This algorithm makes use of an iterative procedure (harmonic product spectrum). We use three iterations.
- (vi) **NNLS.** Mauch introduced an approximate transcription step based on a *Non-Negative Least Squares* algorithm [14]. The resulting chroma features led to a considerable boost of chord recognition performance. The code is published as a “Vamp” plugin.⁴

We compute the initial chroma features with a resolution of 10 Hz. In order to eliminate the influence of dynamics, we normalize to the ℓ^1 norm so that

$$\|c\|_1 = \sum_{n=0}^{N-1} |c_n| = 1. \quad (1)$$

2.2 Key Detection Algorithms

For automatic key detection, we compare four approaches that have been tested successfully on classical music data.

- (1) **Template matching.** For this standard method, the distance between a chroma histogram and a key profile is computed for each of the 24 keys. The profile minimizing the distance gives the global key [28].
- (2) **Profile learning.** Van de Par et al. improved the profile matching algorithm by using a learning procedure for the key profiles [30]. Furthermore, they emphasize the beginning and ending section of the pieces. We extend this idea by separately weighting beginning and ending section. Therefore, we introduce new parameters β and γ to emphasize the beginning and ending, respectively—along with the parameter α from [30].
- (3) **Symmetry model.** Another class of key finding algorithms makes use of geometrical pitch models [2, 5]. We use the symmetry model by Gatzsche and Mehnert that was evaluated for key detection in [16].
- (4) **Final chord.** The algorithm proposed in [31] considers the final chord to estimate the tonic note of the global key—combined with a profile matching for estimating the mode. This algorithm was tested on three datasets of classical music.

⁴ <http://isophonics.net/npls-chroma>

2.3 Classification Features

The basic idea of this paper is to directly use chroma histograms for classification of music styles. We therefore sum up the M chroma vectors c^1, \dots, c^M of a piece in order to obtain a ℓ_1 normalized chroma histogram \mathbf{h} :

$$\hat{\mathbf{h}} = \sum_{i=1}^M c^i, \quad \mathbf{h} = \hat{\mathbf{h}} / \|\hat{\mathbf{h}}\|_1 \quad (2)$$

In order to compare the impact of the chroma computation method, we use four different chroma algorithms from the ones presented in Section 2.1: CP, CLP, EPCP, and NNLS.

As the main contribution of our work, we want to evaluate the relevance of key information for classification. To this end, we test different combinations of key estimation and classification algorithms. Using 3-fold cross-validation, we randomly split our dataset into a training fold (2/3) and a test fold (1/3). For the training stage, the ground truth key annotations are used to split up the data into pieces in major and minor modes. With the same key information, we circularly rotate the chroma histograms so that the tonic note is on the first position:

$$h_k^{\text{rotated}} = h_{(k-k^*) \bmod 12}, \quad (3)$$

with $k \in [0 : 11]$ and k^* denoting the chroma index of the tonic note ($k^* = 0$ for C, etc.). For testing, we use one of the four automatic key detection algorithms presented in Section 2.2. With this key information, we split up the test data according to the mode and again rotate each chroma histogram with respect to the tonic note. The full processing chain of our approach is shown in Figure 2.

To compare against existing methods, we use other types of chroma-based classification features. In [32], a set of *template-based features* for estimating the occurrence of interval and chord types has been proposed. To this end, chroma features are smoothed to different temporal resolutions followed by a multiplication of chroma values according to interval and chord templates. Another group—*tonal complexity features*—makes use of statistical measures on the chroma distribution in order to estimate the tonal complexity of the music on different time scales [33].

3. EVALUATION

In order to estimate the classification performance on unseen data, we apply a two-step evaluation strategy. First, we test the key detection performance of the four methods presented in Section 2.2 and optimize the algorithms’ free parameters (Section 3.2). Second, we perform classification experiments on a different dataset using a Random Forest classifier with chroma histograms as input features. We train separate models for major and minor pieces, respectively. For estimating the importance of the algorithm’s elements, we conduct several baseline experiments.

3.1 Datasets

In our studies, we make use of different datasets. To evaluate key detection performance and optimize param-

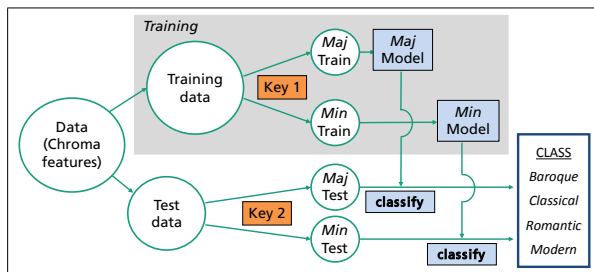


Figure 2. Flow diagram for the classification procedure. For performing cross-validation, the data is split up into training and test set. Each set is sorted with respect to the mode by using different key algorithms or ground truth key annotations (key 1 / key 2), respectively. The trained models for major and minor keys are then used to classify the respective test data.

eters, we use three datasets of classical music recordings with corresponding key annotations. This data has been used for evaluating key detection in published work [23, 30, 31]. The first set (*Symph*) comprises classical and romantic symphonies—each with all movements—from 11 composers containing **115 tracks** in total. The second one—a selection from *Saarland Music Data Western Music (SMD)* [20]—includes music for solo instruments, orchestra, and chamber music. The key annotations for **126 selected tracks** that show clear tonality are available on the corresponding website.⁵ Third, we compiled a dataset of piano music recordings (*Pno*) used for key detection in [23, 30]. This data comprises **237 piano pieces** (Bach, Brahms, Chopin and Shostakovich). We consider these datasets as training data for the key detection step, thus justifying the overfitting procedure for the parameters.

For the classification experiments, we make use of another dataset (*Cross-Era*) containing **1600 audio recordings** of classical music as used in [32, 33]. The data is balanced with respect to the historical periods (each 400 tracks for the Baroque, Classical, Romantic, and Modern period) and instrumentation (200 piano pieces and 200 orchestral pieces per class). We collected expert annotations for the key of 1200 tracks. The modern class has not been considered due to a high amount of atonal pieces. For atonal pieces, we assume little influence of key detection on classification with chroma histograms.⁶ The data is not balanced with respect to the key or the mode (major/minor). We show the key distribution in Figure 3.

3.2 Key Detection Experiments

For estimating the optimal parameters, we run each algorithm with different parameter settings in a stepwise fashion. To that end, we optimize each parameter by maximizing the weighted total performance Λ_t

$$\Lambda_t = (115 \Lambda_{\text{Symph}} + 126 \Lambda_{\text{SMD}} + 237 \Lambda_{\text{Pno}}) / 478 \quad (4)$$

⁵ <http://www.mpi-inf.mpg.de/resources/SMD>

⁶ For example, a dodecaphonic piece of music shows nearly equal pitch class distribution. Thus, its chroma distribution is practically invariant to cyclic shifts.

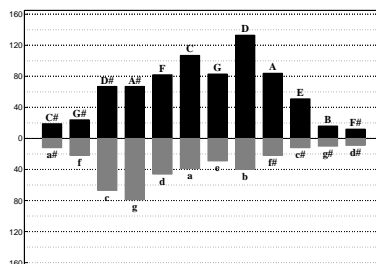


Figure 3. Key distribution (annotations) of the periods Baroque, Classical, and Romantic (1200 pieces) in the dataset *Cross-Era*. Major keys are shown in black and upward direction, minor keys are in grey downwards. The tonic notes are arranged according to the circle of fifths.

and fix the remaining parameters to default or best fit values. For the basic chroma features, we test the six types presented in Section 2.1. We obtain the following results for the different algorithms:

- (1) **Template matching.** We test three pairs (maj / min) of profiles proposed by Krumhansl [12], Temperley [28], and Gomez [7]. In our study, the latter ones performed best. Though these profiles have been developed in combination with HPCP features, NNLS features outperformed these features (84.7 %) followed by CLP.
- (2) **Profile learning.** For the profile training, we performed a cross-validation with 98 % training data, 2 % test data, and 5000 repetitions following [30]. We found best performance for CLP chroma features (92.3 %)—closely followed by NNLS—together with parameters $\alpha = 2$, $\beta = 1$, and $\gamma = 0.25$. We could not reach the result presented in [30] (98 % on the *Pno* dataset). As a reason for this, we assume that the specific chroma features presented in that work (including a masking model) provide additional benefits.

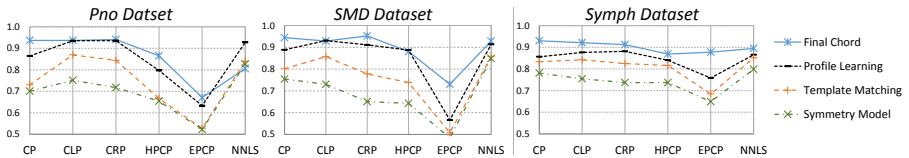


Figure 4. Evaluation of different key detection algorithms. Here, we show the individual key recognition accuracies for the three datasets of classical music. As the basic feature, we compare six types of chroma features.

- (3) **Symmetry model.** This algorithm worked best in conjunction with NMLS chroma. The optimal pitch set energy threshold was found at $f_{TR} = 0.12$. The angular vector value came out best at $w_{sym} = 0.53$ leading to a total performance of 82.6%.
- (4) **Final chord.** The final chord algorithm obtained optimal results on the basis of CP chroma features. For the parameters, $N = 19$ final frames, a root-scale weight exponent of $s = 0.9$, an energy threshold of $f_e = 0.19\%$, and the weight exponents template $m^{(2)}$ have come out best (93.7% accuracy).

The overall results for the key detection evaluation are shown in Figure 4 for the individual datasets. All algorithms considerably depend on the chroma extraction method—especially when the data includes piano music (*Pno*, *SMD*). NMLS features often obtained the best results and seem to be the most stable basis for key detection methods. EPCP features are not a good choice for this purpose. The profile learning and the final chord algorithms performed similarly. Hereby, the first one is rather data-dependent whereas the final chord algorithms requires a fine parameter tuning. In the following, we use the final chord algorithm that showed a slightly better total rate (93.7%) compared to the profile training method (92.3%).

Finally, we test the four key detection methods on a subset of the *Cross-Era* dataset (Section 3.1) using 1200 tracks with key annotations. For each method, we use the feature and parameter setting performing best in the previous experiments.⁷ We obtain a performance of **83.9%** for the template matching algorithm (1), **87.1%** for the profile learning (2), **80.4%** for the symmetry model (3), and **85.4%** for the final chord based method (4). Compared to the optimization datasets, the performance is worse and the differences between the methods are smaller. Profile learning and final chord stay with best results. However, the learning strategy (2) seems to be more robust than the parameter-dependent final chord algorithm (4).

3.3 Classification Experiments

By using the method and parameters performing best in Section 3.2, we now test the influence of key detection on automatic style classification based on the *Cross-Era* dataset. We use a Random Forest (RF) classifier. In order to avoid problems due to the *curse of dimensionality*,

⁷ For the profile learning approach, the profiles are also trained on the previously used datasets *Symph*, *SMD*, and *Pno*.

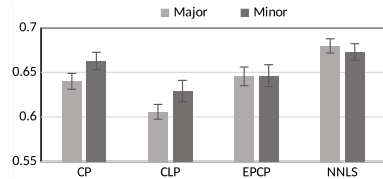


Figure 5. Classification accuracies for different types of chroma features for classification, four classes, $key 1 = key 2 =$ final chord. Bars and error marks indicate mean and standard deviation over 100 initializations of the cross-validation. Here, we do not use LDA (only twelve-dimensional features).

we transform the feature space using Linear Discriminant Analysis (LDA) with three output dimensions. For evaluation, we conduct a 3-fold cross-validation. We use the chroma histograms over the full piece as classification features. As our basic idea, we rotate the chroma histograms to the tonic note (Section 2.3). In the ideal setting, we use the ground truth key annotations for the training data ($key 1$). For the test data ($key 2$), we use the automatically detected key from the final chord algorithm (see Section 3.2).

Major and minor keys exhibit very different tonal structures resulting in distinct typical chroma distributions. The mode-related properties in the chroma distribution may heavily overlay the more subtle differences originating from style. We therefore split up the data into major and minor pieces by using key annotations (training set, $key 1$) or automatic key detection (test set, $key 2$), respectively. On the resulting training data sets, we train separate classification models for major and minor keys. The test data is then classified into style periods using the appropriate classifier model. This procedure is visualized in Figure 2. We then repeat the classification by using the next fold as test data. The whole cross-validation is performed 100 times with new random initialization of the folds.

First, we test the influence of the specific chroma feature implementation on the classification performance. In this experiment, we use the automatic key (final chord algorithm) for both training and test. The results are shown in Figure 5. Classification performance considerably depends on the chroma type. Here, logarithmic compression (CLP)—enhancing weak components—does not improve classification performance. CP and EPCP features perform

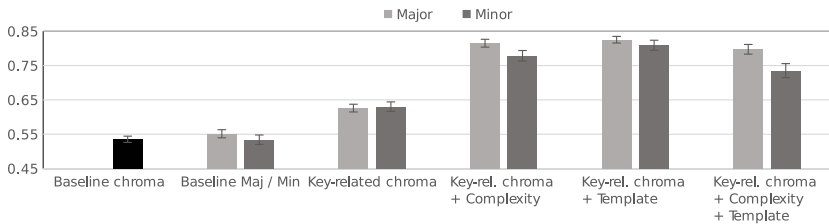


Figure 7. Classification accuracies for different combinations of chroma-based features, four classes, NNLS features. The varying dimensionality of the feature collections is reduced to three dimensions by using LDA.

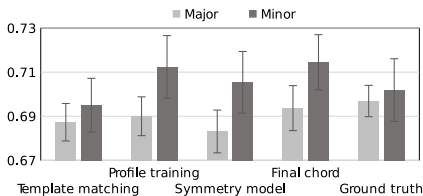


Figure 6. Classification accuracies based on different key detection methods (for *key 1* and *key 2*), three classes, NNLS features. Here, we do not use LDA transformation.

Table 1. Classification results for different key method combinations, three classes, NNLS features.

<i>Key 1</i>	<i>Key 2</i>	<i>Major</i>	<i>Minor</i>
Ground truth	Final chord	70.1%	66.7%
Final chord	Final chord	69.4%	71.5%

similar whereas NNLS features outperform the others by several percentage points. We therefore use NNLS chroma features for the remaining experiments.

Next, we evaluate the dependence of the key-related chroma features on the performance of the automatic key detection. To this end, we once use each of the four methods from Section 3.2 both for training and test data. Since we have no ground truth key annotations for the modern era, we just perform classification of the remaining three classes (1200 pieces). Classification results are similar with all key methods (Figure 6). For profile learning and final chord key detection, the results partly outperform the classification based on ground truth key annotations. We conclude that some of the errors in key detection may have beneficial effects on classification performance. Comparing the classification results with the key detection performance on *Cross-Era*, we find similar behaviour. Thus, a good key detection leads to better classification, sometimes outperforming the use of ground truth key annotations. When using ground truth key for training (*key 1*) and an automatic method for testing (*key 2*), performance values change but do not generally increase (Table 3.3).

In the last study, we compare different types of classification features (Figure 7). For the *baseline chroma*

experiment, we do not use any key information but use the original (absolute) NNLS histograms as classification features—without Major/Minor discrimination (one model for all). *Baseline Maj/Min* makes use of ground truth key annotations for mode selection. This does not lead to increased classification results. For the *key-related chroma* method, we use NNLS rotated with respect to the final chord key, for training and test.⁸ The use of key detection boosts classification results by almost 10%. Next, we combine the key-related chroma histograms with other chroma-based features such as tonal *complexity* or *template*-based features (Section 2.3) leading to improvements of almost 20%. Combining all three types of features does not further increase classification accuracies.

When comparing our results with the outcome of [32, 33], we do not obtain a general performance boost through adding key-related chroma features. Both complexity [33] and template features [32] alone performed similar in the respective experiments—compared to combining them with our features. However, we already obtain remarkable results with key-related features only. These features can be computed with a high computational efficiency.⁹ As the main difference, complexity and template features capture local properties whereas global chroma histograms do not.

4. CONCLUSION

We evaluated four automatic key detection methods and optimized their parameters using three datasets of classical music. On a separate dataset, we performed style classification experiments using key-related chroma histograms as classification features. With such features, the use of an efficient key detection algorithm improves classification accuracy. Thus, automatic key detection constitutes a useful step for such music classification systems. However, involving local chroma-based features leads to a better performance than only using global chroma histograms.

Acknowledgments: C.W. has been supported by the Foundation of German Business (Stiftung der Deutschen Wirtschaft). He thanks Daniel Gärtner for fruitful discussions and Judith Wolff for contributing to key annotations.

⁸ The difference between this result and the NNLS performance in Figure 5 is due to using LDA transformation here.

⁹ Since we only use global chroma, a very coarse time resolution for the time-frequency transform could be applied.

5. REFERENCES

- [1] Eric Backer and Peter van Kranenburg. On Musical Stylemetry: A Pattern Recognition Approach. *Pattern Recognition Letters*, 26(3):299–309, 2005.
- [2] Ching-Hua Chuan and Elaine Chew. Polyphonic Audio Key Finding Using the Spiral Array CEG Algorithm. In *Proceedings of the IEEE International Conference on Multimedia and Expo*, 2005.
- [3] Roger B. Dannenberg, Belinda Thom, and David Watson. A Machine Learning Approach to Musical Style Recognition. In *Proceedings of the International Computer Music Conference (ICMC)*, 1997.
- [4] Ofer Dor and Yoram Reich. An Evaluation of Musical Score Characteristics for Automatic Classification of Composers. *Computer Music Journal*, 35(3):86–97, 2011.
- [5] Gabriel Gatzsche, Markus Mehnert, David Gatzsche, and Karlheinz Brandenburg. A Symmetry Based Approach for Musical Tonality Analysis. In *Proceedings of the 8th International Society for Music Information Retrieval Conference (ISMIR)*, pages 207–210, 2007.
- [6] Irving Godt. Style Periods of Music History Considered Analytically. *College Music Symposium*, 24, 1984.
- [7] Emilia Gómez. *Tonal Description of Music Audio Signals*. PhD thesis, Universitat Pompeu Fabra, Barcelona, 2006.
- [8] Aline Honingh and Rens Bod. Pitch Class Set Categories as Analysis Tools for Degrees of Tonality. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 459–464, 2010.
- [9] Nanzhu Jiang, Peter Grosche, Verena Konz, and Meinard Müller. Analyzing Chroma Feature Types for Automated Chord Recognition. In *Proceedings of the 42nd AES International Conference on Semantic Audio*, pages 285–294, 2011.
- [10] Francis J. Kiernan. Score-based Style Recognition Using Artificial Neural Networks. In *Proceedings of the 1st International Symposium on Music Information Retrieval (ISMIR)*, 2000.
- [11] Peter van Kranenburg. Composer Attribution by Quantifying Compositional Strategies. In *Proceedings of the 7th International Society for Music Information Retrieval Conference (ISMIR)*, 2006.
- [12] Carol L. Krumhansl. *Cognitive Foundations of Musical Pitch*. Oxford Psychology Series. Oxford University Press, 1990.
- [13] Kyogu Lee. Automatic Chord Recognition from Audio Using Enhanced Pitch Class Profile. In *Proceedings of the International Computer Music Conference (ICMC)*, 2006.
- [14] Matthias Mauch and Simon Dixon. Approximate Note Transcription for the Improved Identification of Difficult Chords. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 135–140, 2010.
- [15] Lesley Mearns and Simon Dixon. Characterisation of Composer Style Using High Level Musical Features. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, 2010.
- [16] Markus Mehnert, Gabriel Gatzsche, and Daniel Arndt. Symmetry Model Based Key Finding. In *Proceedings of the 126th AES Convention*, 2009.
- [17] Meinard Müller and Sebastian Ewert. Towards Timbre-Invariant Audio Features for Harmony-Based Music. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):649–662, 2010.
- [18] Meinard Müller and Sebastian Ewert. Chroma Toolbox: MATLAB Implementations for Extracting Variants of Chroma-Based Audio Features. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 215–220, 2011.
- [19] Meinard Müller, Frank Kurth, and Michael Clausen. Chroma-Based Statistical Audio Features for Audio Matching. In *Proceedings Workshop on Applications of Signal Processing (WASPAA)*, pages 275–278, 2005.
- [20] Meinard Müller, Verena Konz, Wolfgang Bogler, and Viora Arifi-Müller. Saarland Music Data. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, 2011.
- [21] Meinard Müller, Frank Kurth, and Michael Clausen. Audio Matching via Chroma-Based Statistical Features. In *Proceedings of the 6th International Society for Music Information Retrieval Conference (ISMIR)*, pages 288–295, 2005.
- [22] Mitsunori Oghihara and Tao Li. N-Gram Chord Profiles for Composer Style Identification. In *Proceedings of the 9th International Society for Music Information Retrieval Conference (ISMIR)*, 2008.
- [23] Steffen Pauws. Musical key extraction from audio. In *Proceedings of the 5th International Society for Music Information Retrieval Conference (ISMIR)*, 2004.
- [24] Carlos Pérez-Sancho, D. Rizo, José Manuel Iñesta, Pedro José Ponce de León, S. Kersten, and Rafael Ramirez. Genre Classification of Music by Tonal Harmony. *Intelligent Data Analysis*, 14(5):533–545, 2010.
- [25] Pedro José Ponce de León and José Manuel Iñesta. A Pattern Recognition Approach For Music Style Identification Using Shallow Statistical Descriptors. *IEEE Transactions on System, Man and Cybernetics - Part C: Applications and Reviews*, 37(2):248–257, 2007.
- [26] Christian Simmermacher, Da Deng, and Stephen Crane. Feature Analysis and Classification of Classical Musical Instruments: An Empirical Study. In *Advances in Data Mining. Applications in Medicine, Web Mining, Marketing, Image and Signal Mining*, volume 4065 of *Lecture Notes in Computer Science*, pages 444–458. Springer, Berlin and Heidelberg, 2006.
- [27] Michael Stein, B. M. Schubert, Matthias Gruhne, Gabriel Gatzsche, and Markus Mehnert. Evaluation and Comparison of Audio Chroma Feature Extraction Methods. In *Proceedings of the 126th AES Convention*, 2009.
- [28] David Temperley. *The Cognition of Basic Musical Structures*. MIT Press, 2001.
- [29] George Tzanetakis and Perry Cook. Musical Genre Classification of Audio Signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.
- [30] Steven van de Par, Martin F. McKinney, and André Redert. Musical Key Extraction From Audio Using Profile Training. In *Proceedings of the 7th International Society for Music Information Retrieval Conference (ISMIR)*, 2006.
- [31] Christof Weiß. Global Key Extraction from Classical Music Audio Recordings Based on the Final Chord. In *Proceedings of the 10th Sound and Music Computing Conference (SMC)*, 2013.
- [32] Christof Weiß, Matthias Mauch, and Simon Dixon. Timbre-Invariant Audio Features for Style Analysis of Classical Music. In *Proceedings of the Joint Conference 40th ICMC and 11th SMC*, 2014.
- [33] Christof Weiß and Meinard Müller. Tonal Complexity Features for Style Classification of Classical Music. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2015.

CHORD DETECTION USING DEEP LEARNING

Xinquan Zhou
Center for Music Technology
Georgia Institute of Technology
royzxq@gmail.com

Alexander Lerch
Center for Music Technology
Georgia Institute of Technology
alexander.lerch@gatech.edu

ABSTRACT

In this paper, we utilize deep learning to learn high-level features for audio chord detection. The learned features, obtained by a deep network in bottleneck architecture, give promising results and outperform state-of-the-art systems. We present and evaluate the results for various methods and configurations, including input pre-processing, a bottleneck architecture, and SVMs vs. HMMs for chord classification.

1. INTRODUCTION

The goal of automatic chord detection is the automatic recognition of the chord progression in a music recording. It is an important task in the analysis of western music and music transcription in general, and it can contribute to applications such as key detection, structural segmentation, music similarity measures, and other semantic analysis tasks. Despite early successes in chord detection by using pitch chroma features [6] and Hidden Markov Models (HMMs) [26], recent attempts at further increasing the detection accuracy are only met with moderate success [4, 28].

In recent years, deep learning approaches have gained significant interest in the machine learning community as a way of building hierarchical representations from large amounts of data. Deep learning has been applied successfully in various fields; for instance, a system for speech recognition utilizing deep learning was able to outperform state-of-the-art systems not using deep learning [10]. Several studies indicate that deep learning methods can be very successful when applied to Music Information Retrieval (MIR) tasks, especially when used for feature learning [1, 9, 13, 16]. Deep learning, with its potential to untangle complicated patterns in a large amount of data, should be well suited for the task of chord detection.

In this work, we investigate Deep Networks (DNs) for learning high-level and more representative features in the context of chord detection, effectively replacing the widely used pitch chroma intermediate representation. We present individual results for different pre-processing options such as time splicing and filtering (see Sect. 3.2), architectures (see Sect. 3.4), and output classifiers (see Sect. 4).



© Xinquan Zhou, Alexander Lerch.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Xinquan Zhou, Alexander Lerch. “Chord Detection Using Deep Learning”, 16th International Society for Music Information Retrieval Conference, 2015.

2. RELATED WORK

During the past decade, deep learning has been considered by the machine learning community to be one of the most interesting and intriguing research topics. Deep architectures promise to remove the necessity of custom-designed and manually selected features as neural networks should be more powerful in disentangling interacting factors and thus be able to create meaningful high-level representations of the input data. Generally speaking, deep learning combines deep neural networks with an unsupervised learning model. Two major learning models are widely used for unsupervised learning: Restricted Boltzmann Machines (RBMs) [11] and Sparse Auto Encoders [24]. A deep architecture comprises multiple stacked layers based on one of these two models. These layers can be trained one by one, a process that is referred to as “pre-training” the network. In this work, we employ RBMs to pre-train the deep architecture in an unsupervised fashion; this is called a Deep Belief Network (DBN) [11]. DBNs, composed of a stack of RBMs, essentially share the same topology with general neural networks: DBNs are generative probabilistic models with one visible layer and several hidden layers.

Since Hinton et al. proposed a fast learning algorithm for DBNs [11], it has been widely used for initializing deep neural networks. In deep structures, each layer learns relationships between units in lower layers. The complexity of the system increases with an increasing number of RBM layers, making the structure—in theory—more powerful. An extra softmax output layer can be added to the top of the network (see Eqn (6)) [18]; its output can be interpreted as the likelihood of each class.

LeCun and Bengio introduced the idea of applying Convolutional Neural Networks (CNNs) to images, speech, and other time-series signals [15]. This approach allows to deal with the variability in time and space to a certain degree, as CNNs can be seen as a special type of neural network in which the weights are shared across the input within a certain spatial or temporal area. The weights thus act as a kernel filter applied to the input. CNNs have been particularly successful in image analysis. For example, Norouzi et al. used Convolutional RBMs to learn shift-invariant features [22].

The results of a network depend largely on the network architecture. For example, Grezl et al. used a so-called bottleneck architecture neural network to obtain features for speech recognition and showed that these features improve the accuracy of the task [8]. The principle behind

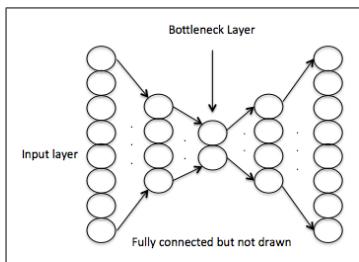


Figure 1. Visualization of a bottleneck architecture

the bottleneck-shaped architecture is that the number of neurons in the middle layer is lower than in the other layers as shown in Fig. 1. A network with bottleneck can be structured in two sections: (i) Section 1 from the first layer to the bottleneck layer, with a gradual decrease of the number of neurons per layer, functions as an encoding or compression process which compacts relevant information and discards redundant information, and (ii) Section 2 from the bottleneck layer to the last layer with a gradual increase in the number of neurons per layer. The function of this part can be interpreted as a decoding process. An additional benefit of bottleneck architectures is that they can reduce overfitting by decreasing the system complexity.

Recently, more researchers investigated deep learning in the context of MIR. Lee et al. pioneered the application of convolutional deep learning for audio feature learning [16]. Hamel et al. used the features learned from music with a DBN for both music genre classification and music auto-tagging [9]; their system was successful in MIREX 2011 with top-ranked results. Battenberg employed a conditional DBN to analyze drum patterns [1]. The use of deep architectures for chord detections, however, has not yet been explored, although modern neural networks have been employed in this field. For instance, Boulanger et al. investigated recurrent neural networks [2] and Humphrey has explored CNNs [12, 14]. While they also used the concept of pre-training, their architectures have only two or 3 layers and thus cannot be called “deep”.

The basic building blocks of most modern approaches to chord detection can be traced back to two seminal publications: Fujishima introduced pitch chroma vectors extracted from the audio as input feature for chord detection [6] and Sheh et al. proposed to use HMMs for representing chords as hidden states and to model the transition probability of chords [26]. Since then, there have been a lot of studies using chroma features and HMMs for chord detection [5, 23]. Examples for recent systems are Ni et al., using a genre-independent chord estimation method based on HMM and chroma features [21] and Cho and Bello, who used multi-band features and a multi-stream HMM for chord recognition [4]. Training HMMs with pitch chroma features arguably is the standard approach for this task and the progress is less marked by major innovations but by

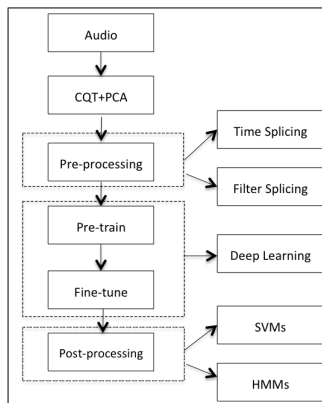


Figure 2. The overview of our system

optimizing and tuning specific components.

3. SYSTEM OVERVIEW

Figure 2 gives an overview of all components and processing steps of the presented system. The following section will discuss all of these steps in detail.

3.1 Input Representation

The input audio is converted to a sample rate of 11.025 kHz. Then, a Constant Q transform (CQT) is applied. The CQT [3] is a perceptually inspired time-frequency transformation for audio. The resulting frequency bins are equally spaced on a logarithmic (“pitch”) scale. It has the advantage of providing a more musically and perceptually meaningful spectral representation than the DFT. We used an implementation of the CQT as a filterbank of Gabor filters, spaced at 36 bins per octave, i.e., 3 bins per semitone, yielding 180 bins representing a frequency range spanning from 110 Hz to 3.520 kHz. Finally, we used Principal Component Analysis (PCA) for decorrelation, and applied Z-Score normalization [27].

3.2 Pre-processing

Neighboring frames of the input representation can be expected to contain similar content, as chords will not change on a frame-by-frame basis. In order to take into account the relationship between the current frame and previous and future frames, we investigate the application of several pre-processing approaches.

3.2.1 Time Splicing

Time splicing is a simple way to extend the current frame with the data of neighboring frames by concatenating the frames into one larger superframe. In first order time splicing, we concatenate the current frame, the previous frame,

and the following frame. Thus, each superframe consists of three neighboring frames. Since the same operation will be applied to all frames, there will be overlap introduced between neighboring superframes.

3.2.2 Convolution

CNNs are extensively used in tasks with highly correlated inputs (e.g., the recognition of hand-written digits). Many time series show similar properties so that CNNs seem to be an appropriate choice in the context of audio, too. Essentially, CNNs have one or more convolutional layers between the input and lower layers of the neural network. The function of a convolutional layer can be interpreted as the application of a linear filter plus a non-linear transformation, sometimes also combined with a pooling operation:

$$Y = \text{pool}(\text{sigm}(K * X + B)), \quad (1)$$

in which Y is the output of a convolutional layer, K is the linear kernel filter (i.e., the impulse response), X is the input, B is the bias, $\text{sigm}()$ is a non-linear transform, and $\text{pool}()$ is a down-sampling operation. The uniqueness of convolutional networks stems from the convolution operation applied to the input X . Since, unfortunately, we had no access to a deep learning toolbox with support for the convolution operation in the time domain, we opted to employ an optional pre-processing step inspired by CNNs, namely by applying filters to the input of the network. However, instead of learning the filters, we evaluate several manually designed filters: a single-pole low pass filter and two FIR low pass filters with exponentially shaped impulse responses. The single pole low pass filter produces the output y for an input x , given the parameter α :

$$y_n = (1 - \alpha)y_{n-1} + \alpha x_n \quad (2)$$

We apply anti-causal filtering and filter the signal in both directions so that the resulting overall filter has a zero-phase response.

The other two low pass filters have exponential decay shaped impulse response. The difference equations are given in Eqn (3) and Eqn (4).

$$y_1(n) = \sum_{k=1}^N a^{-k+1} x(n - N + k) \quad (3)$$

$$y_2(n) = \sum_{k=1}^N a^{-k+1} x(n + N - k) \quad (4)$$

The filter length is N and a is the exponential base. These two filters are not centered around the current frame anymore but shifted by N frames. Their impulse responses are symmetric to each other. One could interpret these filters as focusing on past and future frames, respectively. The presented filters will be referred to as “extension filters”.

The ideas of splicing and convolution can be combined, as exemplified in Fig. 3.

Furthermore, similar to the process in CNNs, a maximum pooling operation on the output of the spliced filters is optionally applied. The operation takes the maximum value among different filters per “bin”.

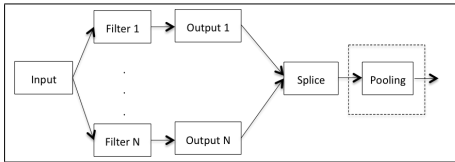


Figure 3. Splicing output of different filters

3.3 Training

It is impractical to train DNNs directly with back propagation using gradient descent due to their deep structure and the limited amount of training samples. Therefore, the network is usually initialized by an unsupervised pre-training step. As our network consists of RBMs, Gibbs sampling can be used for training [11]. The objective is to retain as much information as possible between input and output.

The computation for layer l can be represented as:

$$Y_l = \text{sigm}(W_l X_l + B_l), \quad (5)$$

which is identical to many traditional neural networks. Thus, a standard back propagation can be applied after pre-training to fine-tune the network in a supervised manner. The loss criterion we use in this work is cross-entropy.

3.4 Architecture

We investigate a deep network with 6 layers in two different architectures. The common architecture features the same amount of neurons in every layer, in our case 1024. The bottleneck architecture has 256 neurons in the middle layer and 512 neurons in the layers neighboring the middle layer. The remaining layers consist 1024 neurons each (compare [8]). A softmax output layer is stacked on top of both architectures as described by Eqn (6).

$$\text{softmax}(Y_l) = \frac{\exp(Y_l)}{\sum_{k=1}^N \exp(Y_k)} \quad (6)$$

The network is implemented using the Kaldi package developed by John Hopkins University [25].

4. CLASSIFICATION

The output of the softmax layer can be interpreted as the likelihood of each chord class; simply taking the maximum will provide a class decision (this method will be referred to as *Argmax*). Alternatively, the output can be treated as intermediate feature vector that can be used as an input to other classifiers for computing the final decision.

4.1 Support Vector Machine

Support Vector Machines (SVMs) are, as widely used classifiers with generally good performance. The SVM is trained using the output of the network as features, and the classification is carried out frame by frame. The classification is followed by a simple prediction smoothing.

4.2 Hidden Markov Model

HMMs are, as pointed out above, the standard classifier for automatic chord detection because the characteristics of the task fit the HMM approach well: Chords are hidden states that can be estimated from observations (feature vectors extracted from the audio signal), and the likelihood of chord transitions can be modeled with transition probabilities. Modified HMMs such as ergodic HMMs and key-independent HMMs have been also explored for this task [17, 23]. In this work we are mostly interested in the performance comparison between high-level features, so a simple first-order HMM is used. Given the probabilistic characteristic of the softmax output layer, it can be directly as emission probabilities for the HMM. Therefore, there is no need to train the HMM using, e.g., the commonly used Baum-Welch algorithm. Instead, the histogram of each class in our training is used as initial probabilities, and the bigram of chord transitions is used to compute the transition probabilities. Finally, we employ the Viterbi decoding algorithm to find the globally optimal chord sequence.

5. EVALUATION PROCEDURE

5.1 Dataset

Our dataset is a combination of several different datasets, yielding a 317-piece collection. The data is composed of

- 180 songs from the *Beatles dataset* [19],
- 100 songs from the *RWC Pop dataset* [7],
- 18 songs from the *Zweieck dataset* [19], and
- 19 songs from *Queen dataset* [19].

The pre-processing as described in Sect. 3.2 ensures identical input audio formats.

5.2 Methodology

The dataset is divided randomly into two parts: 80% for the training set and 20% for the test set. On the training scale, we use a frame-based strategy, which means we divide each song into frames, and treat each frame as an independent training sample. On average each song is divided into about 1200 frames resulting in approximately 300k training samples and approximately 76k test samples.

Within the training set, 10% of the data is used as a validation set. For the post-processing, all data in the training set will be used to train the post-classifier.

Time constraints and the workload requirements for training deep networks made a cross validation for evaluation impractical.

The chosen ground truth for classification are major and minor triads for every root note, resulting in a dictionary of $24 + 1$ chord labels. Ground truth time-aligned chord symbols are mapped to this major/minor dictionary:

$$\text{Chord}_{majmin} \subset \{N\} \cup \{S \times maj, min\} \quad (7)$$

with S representing the 12 pitch classes (root notes) and N being the label for unknown chords. In the calculation of the detection accuracy, the following chord types are mapped to the corresponding major/minor in the dictionary:

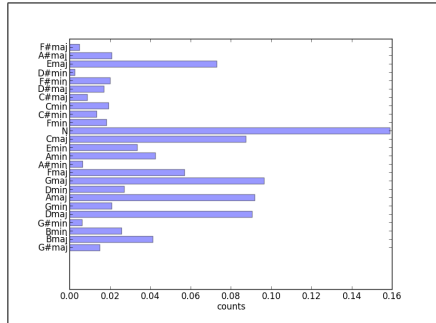


Figure 4. Chords histogram

triad major/minor and seventh major/minor. Other chord types are treated as unknown chords. For instance, G:maj and G:maj7 are mapped to ‘G:maj’; G:dim and G:6 are all mapped to ‘N’. The histogram of chords in our dataset after such mapping is shown in Fig. 4.

5.3 Evaluation Metric

The used evaluation metric is the same as proposed in the audio chord detection task for MIREX 2013: the Weighted Chord Symbol Recall (WCSR). WCSR is defined as the total duration of segments with correct prediction as formulated in Eqn (8):

$$WCSR = \frac{1}{N} \sum_{k=1}^n C_k, \quad (8)$$

in which n is the number of test samples (songs), N is the total number of frames in all test samples, and C_k is the number of frames that are correctly detected in the k th sample.

6. EXPERIMENTS

6.1 Post-classifiers

In this experiment, the network is initialized with pre-training, followed by fine tuning using back propagation. This configuration will be referred to as $DN_{DBN-DNN}$. No pre-processing is applied to the data; the input is simply the input representation (CQT followed by PCA) as described in Sect. 3.1. The chosen architecture is the bottleneck architecture. Three different classifiers are compared: the maximum of the softmax output (Argmax), an SVM, and an HMM.

The results listed in Table 1 are unambiguous and unsurprising: the HMM with Viterbi decoding outperforms the SVM; using HMMs with a model for transition probabilities is an appropriate approach to chord detection as it models the dynamic properties of chord progressions, which cannot be done with non-dynamic classifiers such as SVMs. One noteworthy result is that the SVM does not

Training Scenario	Classifier	WCSR
$DN_{DBN-DNN}$	Argmax()	0.648
$DN_{DBN-DNN}$	SVM	0.645
$DN_{DBN-DNN}$	HMM	0.755

Table 1. Chord detection performance using different post-classifiers

α	Pre-processing	WCSR
0.25	Filtering	0.758
0.25	Spliced Filters	0.912
0.5	Filtering	0.787
0.5	Spliced Filters	0.857
0.75	Filtering	0.798
0.75	Spliced Filters	0.919

Table 2. Chord detection performance using different filter parameters

improve the WCSR compared to the direct (Argmax) output of the network. Apparently, the SVM is not able to improve separability of the learned output features.

6.2 Pre-processing

As stated in Sect. 3.2, we are interested in the application of different filters in the pre-processing stage. In the first experiment (*Filtering*), an anti-causal single pole filter (see Eqn (2)) is evaluated with the parameter α set to 0.25, 0.5, and 0.75, respectively. The second experiment (*Spliced Filters*), splices these filter outputs with the outputs of the extension filters as introduced in Sect. 3.2. These experiments are carried out with the $DN_{DBN-DNN}$ training scenario, a bottleneck architecture, and an HMM classifier. Table 2 lists the results of these pre-processing variants. It can be observed that the network trained with filtered inputs slightly outperforms the network without pre-processing; splicing the filtered input with the extension filter outputs increases the results drastically.

6.3 Architecture

6.3.1 Common vs. Bottleneck

The results of Grezl et al. indicate that a bottleneck architecture should be more suitable to learn high-level features than a common architecture and reduce overfitting [8]. In order to verify these characteristics for our task, the performance of both architectures is evaluated in comparison. The results are listed in Table 3 for three pre-processing scenarios: no additional pre-processing (*None*), *Spliced Filters* and spliced filters followed by a max pooling (*Pooling*). In order to allow conclusions about overfitting, both the WCSR of the test set and the training set are reported. All results are computed for the $DN_{DBN-DNN}$ training scenario with HMM classifiers.

The results show that the bottleneck architecture gives significantly better results ($p = 0.023$) on the test set

Architecture	Pre-processing	Training WCSR	WCSR
Common	None	0.843	0.703
Bottleneck	None	0.855	0.755
Common	Spliced Filters	0.985	0.876
Bottleneck	Spliced Filters	0.936	0.919
Common	Pooling	0.965	0.875
Bottleneck	Pooling	0.960	0.916

Table 3. Chord detection performance for different architectures and pre-processing steps

Learning Targets	WCSR
Single-Label — 25 Chord Classes	0.919
Multi-Label — 12 Pitch Classes	0.78

Table 4. Chord detection performance for single-label vs. multi-label learning

(WCSR). Note that this is not true for the training set (*Training WCSR*), for which the common architecture achieves results in the same range or better than the bottleneck architecture. The difference between the results on the training set and the test set are thus much larger for the common architecture than for the bottleneck architecture. The bottleneck architecture is clearly advantageous to use in this task: it reduces complexity and thus the training workload and increases the classification performance significantly. Furthermore, the comparison of classifier performance between training and test set in Table 3 clearly indicates that the common architecture tends to fit more to the training data, and is thus more prone to overfitting.

6.3.2 Single-Label vs. Multi-Label

As mentioned above, the pitch chroma is the standard feature representation for audio chord detection. Since we use the output of our deep network as feature, it seems an intuitive choice to learn pitch class information (and thus, a pitch chroma) instead of the chord classes. By doing so, the number of outputs is reduced by a factor of two (or higher in the case of more chords), and there would also be a closer relation between the output and the input representation, the CQT. Therefore, the abstraction and complexity of the task might be decreased. It will, however, lead to another issue: the single-label output (one chord per output) will be changed into a multi-label output (multiple pitches per output). Therefore, the learning has to be modified to allow multiple simultaneous (pitch class) labels. The experiment is carried out with both Splicing and Filtering in the pre-processing, the $DN_{DBN-DNN}$ training scenario, and HMM classifiers. Table 4 lists the results.

Boulangier-Lewandowski et al. report combining chroma features with chord labels for their recurrent neural network and report a slightly improved result [2]. They do not, however, provide a detailed description of this combination. As can be seen from the table, the result for multi-label training is clearly lower than the result for single-label training.

Method	WCSR
Chordino	0.625
Best Configuration	0.919
Best Configuration with Max Pooling	0.916

Table 5. Comparison of the performance of the best configuration with Chordino

Possible reasons for bad performance include (i) difficulties with multi-target learning, since it increases the difficulty to train; furthermore, our implementation of multi-label training might be sub-optimal as the same posterior is assigned to each target without any information on the pitch class energy, and (ii) the issue that not all pitches always sound simultaneously in a chord (or might be missing altogether) might have larger impact on the multi-label training than on the single-label training.

6.4 Results & Discussion

It is challenging to compare the results to previously published results due to varying evaluation methodologies, metrics, and datasets. It seems that the results of Cho and Bello [4], who reported a performance of about 76%, were computed with a comparable dataset. The recent MIREX results on Chord Detection generally show lower accuracy but use a different evaluation vocabulary. In order to provide a baseline result to put results into perspective, we present the results of Chordino [20] with the default settings, computed on our dataset. It should be pointed out that this comparison is unfair as Chordino is able to detect as many as 120 chords, compared to our 24. The label mapping strategies are another significant issue for Chordino. Our label mapping results in nearly sixth of the total label being “N”, which might have negative impact on the Chordino results. The Chordino results are mapped to major/minor the same way as the ground truth annotations. The results are shown in Table 5. In the table, the *Best Configuration* is using Bottleneck architecture, spliced filters ($\alpha = 0.75$) as preprocessing, single label learning targets, and Viterbi decoding as post-classifier. The *Best Configuration with Max Pooling* is the same as the best configuration except applying another max pooling layer after the spliced filters. The latter configuration has a much reduced computational workload. The presented results are clearly competitive with existing state-of-the-art systems.

7. CONCLUSION & FUTURE WORK

In this work, we presented a system which applies deep learning to the MIR task of automatic chord detection. Our model is able to learn high-level probabilistic representations for chords across various configurations. We have shown that the use of a bottleneck architecture is advantageous as it reduces overfitting and increases classifier performance, and that the choice of appropriate input filtering and splicing can significantly increase classifier performance.

Learning a pitch class vector instead of chord likelihood

by incorporating multi-label learning proved to be less successful. The idea has, however, a certain appeal and would allow the number of output nodes to be independent of the number of chords to be detected. It is also conceivable to investigate a different option for the network output: instead of training chords or pitch classes we could — under the assumption that we are only after chords comprised of stacked third intervals — train the output with octave-independent third intervals in a multi-label scenario with 24 output nodes.

8. REFERENCES

- [1] Eric Battenberg and David Wessel. Analyzing drum patterns using conditional deep belief networks. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 37–42, 2012.
- [2] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Audio chord recognition with recurrent neural networks. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 335–340, 2013.
- [3] Judith C Brown. Calculation of a constant q spectral transform. *The Journal of the Acoustical Society of America*, 89(1):425–434, 1991.
- [4] Taemin Cho and Juan P Bello. Mirex 2013: Large vocabulary chord recognition system using multi-band features and a multi-stream HMM. *Music Information Retrieval Evaluation eXchange (MIREX)*, 2013.
- [5] Taemin Cho, Ron J Weiss, and Juan Pablo Bello. Exploring common variations in state of the art chord recognition systems. In *Proceedings of the Sound and Music Computing Conference (SMC)*, pages 1–8, 2010.
- [6] Takuya Fujishima. Realtime chord recognition of musical sound: A system using common lisp music. In *Proceedings of the International Computer Music Conference (ICMC)*, volume 1999, pages 464–467, 1999.
- [7] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. Rwc music database: Popular, classical and jazz music databases. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, volume 2, pages 287–288, 2002.
- [8] Frantisek Grezl, Martin Karafiát, Stanislav Kontár, and J Cernocky. Probabilistic and bottle-neck features for lvcsr of meetings. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages IV–757. IEEE, 2007.
- [9] Philippe Hamel and Douglas Eck. Learning features from music audio with deep belief networks. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 339–344. Utrecht, The Netherlands, 2010.

- [10] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *Signal Processing Magazine, IEEE*, 29(6):82–97, 2012.
- [11] Geoffrey Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.
- [12] Eric J Humphrey and Juan Pablo Bello. Rethinking automatic chord recognition with convolutional neural networks. In *Proceedings of the International Conference on Machine Learning and Applications (ICMLA)*, volume 2, pages 357–362. IEEE, 2012.
- [13] Eric J Humphrey, Juan Pablo Bello, and Yann LeCun. Moving beyond feature design: Deep architectures and automatic feature learning in music informatics. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 403–408, 2012.
- [14] Eric J Humphrey, Taemin Cho, and Juan Pablo Bello. Learning a robust tonnetz-space transform for automatic chord recognition. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 453–456. IEEE, 2012.
- [15] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361:310, 1995.
- [16] Honglak Lee, Peter Pham, Yan LARGMAN, and Andrew Y Ng. Unsupervised feature learning for audio classification using convolutional deep belief networks. In *Advances in neural information processing systems*, 2009.
- [17] Kyogu Lee and Malcolm Slaney. Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):291–301, 2008.
- [18] Thomas M Martinetz, Stanislav G Berkovich, and Klaus J Schulten. Neural-gas’ network for vector quantization and its application to time-series prediction. *Neural Networks, IEEE Transactions on*, 4(4):558–569, 1993.
- [19] Matthias Mauch, Chris Cannam, Matthew Davies, Simon Dixon, Christopher Harte, Sefki Kolozali, Dan Tidhar, and Mark Sandler. Omras2 metadata project 2009. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, 2009.
- [20] Matthias Mauch and Simon Dixon. Approximate note transcription for the improved identification of difficult chords. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 135–140, 2010.
- [21] Yizhao Ni, Matt McVicar, Raul Santos-Rodriguez, and Tijl De Bie. Using hyper-genre training to explore genre information for automatic chord estimation. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 109–114, 2012.
- [22] Mohammad Norouzi, Mani Ranjbar, and Greg Mori. Stacks of convolutional restricted boltzmann machines for shift-invariant feature learning. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2735–2742. IEEE, 2009.
- [23] Hélene Papadopoulos and Geoffroy Peeters. Large-scale study of chord estimation algorithms based on chroma representation and hmm. In *Proceedings of the International Workshop on Content-Based Multimedia Indexing (CBMI)*, pages 53–60. IEEE, 2007.
- [24] Christopher Poultney, Sumit Chopra, Yann L Cun, et al. Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems*, pages 1137–1144, 2006.
- [25] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hanemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, Jan Silovsky, Georg Stemmer, and Karel Vesely. The kaldi speech recognition toolkit. In *Proceedings of the Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, December 2011. IEEE Catalog No.: CFP11SRW-USB.
- [26] Alexander Sheh and Daniel PW Ellis. Chord segmentation and recognition using em-trained hidden markov models. *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 185–191, 2003.
- [27] J Sola and J Sevilla. Importance of input data normalization for the application of neural networks to complex industrial problems. *Nuclear Science, IEEE Transactions on*, 44(3):1464–1468, 1997.
- [28] Yushi Ueda, Yuuki Uchiyama, Takuya Nishimoto, Nobutaka Ono, and Shigeki Sagayama. HMM-based approach for automatic chord detection using refined acoustic features. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5518–5521. IEEE, 2010.

TEMPORAL MUSIC CONTEXT IDENTIFICATION WITH USER LISTENING DATA

Cameron Summers

Gracenote

csummers@gracenote.com

Phillip Popp

Gracenote

ppopp@gracenote.com

ABSTRACT

The times when music is played can indicate context for listeners. From the peaceful song for waking up each morning to the traditional song for celebrating a holiday to an up-beat song for enjoying the summer, the relationship between the music and the temporal context is clearly important. For music search and recommendation systems, an understanding of these relationships provides a richer environment to discover and listen. But with the large number of tracks available in music catalogues today, manually labeling track-temporal context associations is difficult, time consuming, and costly.

This paper examines track-day contexts with the purpose of identifying relationships with specific music tracks. Improvements are made to an existing method for classifying Christmas tracks and a generalization to the approach is shown that allows automated discovery of music for any day of the year. Analyzing the top 50 tracks obtained from this method for three well-known holidays, Halloween, Saint Patrick's Day, and July 4th, precision@50 was 95%, 99%, and 73%, respectively.

1. INTRODUCTION

With the ever increasing amount of recorded music, structured metadata is important to organize it. For holiday music, there is some metadata that indicates an association with a music track, often Christmas [1], but comprehensive labeling for other holidays is still lacking. One reason for this is the varying nature of holiday music. Across geographies, cultures, and time, what music is used to celebrate holidays changes dramatically. There is a bit of a paradox as to whether a holiday track is so because the artist recorded it for that purpose or the listeners use it to celebrate¹. Given this complex landscape of holiday music, manual labeling of a large number of music tracks is difficult, time consuming, and costly. Methods for automated labeling are desirable for large scale organization, further

¹ The interpretation of the authors of what is truly holiday music is the latter.

improving the capabilities of music search and recommendation systems.

One automated approach is using text search of track names or album names for keywords also associated with the target holiday [2]. For example, tracks with the keywords "winter" or "spooky" may be likely associated with Christmas or Halloween, respectively. This approach has drawbacks, however. First, it requires experts to create keywords lists, which can be costly or difficult, particularly for music in different languages. Second, the keywords do not guarantee correct track-holiday association, particularly for ambiguous words like "whiskey", which could be linked contextually to Saint Patrick's Day or simply drinking beverages. This problem is compounded when using multiple keywords, as is required for a comprehensive set of tracks.

Another automated approach for labeling holiday music is through user crowdsourcing. LastFM (last.fm), for example, allows users to add their own tags to music tracks, which include tags for some holidays like Halloween [3]. This has the advantages of outsourcing the work of labeling and getting a better representation of the holiday music preferences of a larger number of listeners. But this too has drawbacks. Users tend to only label popular tracks and artists, leading to imbalanced coverage. The quality of these tags can suffer to due misspellings, synonyms, biases, or dishonest labeling. And the users providing tags are still typically a small subset of the total users of a service [4].

Alternatively, leveraging user listening data avoids the quality issues associated with user tagging and keyword association, can utilize the entire user base, and is language agnostic. Researchers have studied temporal dynamics of user data previously to understand context. [5] examined temporal context to improve biosurveillance. [6] and [7] classified web search queries using features in the popularity signal over time and in music, [8], [9], and [10] show the usefulness of temporal analysis in recommendation systems. An approach proposed by [11] exploits user listening data to automatically label tracks as associated with Christmas. However, the approach performs poorly for other holidays in our experiments. In this paper we show that the methodology in [11] can be improved and generalized to discover tracks associated with other holidays throughout the year.



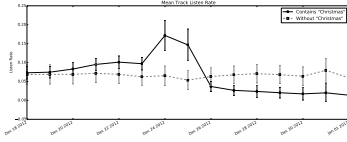


Figure 1. Mean listen rates R for tracks above 1,500 total listen threshold with "Christmas" in track or album name (solid line) and tracks without (dotted line) for December 18, 2012 - January 1, 2013.

2. METHODOLOGY

[11] hypothesized that the listening signals of two tracks, one associated with a holiday and one not, will have differing and detectable patterns on and around that holiday. This can readily be seen for Christmas tracks and non-Christmas tracks in Figure 1. In this section we show the methodology in [11] for detecting Christmas tracks and propose improvements.

2.1 Listening Rates

The form of the raw data is listening events in which a known user has listened to a known track at a date and time. If a track is associated with a day, we expect users to engage more relative to other time periods. The signal used in [11] can be described as user engagement,

$$E_{ij} = \sum_{k=1}^U c_{ijk} \quad (1)$$

which is the total number of listens for all users for track i in time period j .

In Eqn (1), c_{ijk} is an element of C , and $C \in \mathbb{R}^{T \times W \times U}$ where T is the number of tracks, W is the number of time periods, and U is the number of users. To account for differences between popularity of tracks, the E was normalized across the periods of time as described by

$$R_{ij} = \frac{E_{ij}}{\sum_{l=1}^W E_{il}} \quad (2)$$

which were the listen rates used to train the Christmas model.

We propose a new signal based on absolute user engagement, \hat{E} . Given the function

$$f(x) = \begin{cases} 1, & \text{if } x > 0 \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

\hat{E}_{ij} is the number of users who listened to a track i in time period j and is calculated by

$$\hat{E}_{ij} = \sum_{k=1}^U f(c_{ijk}). \quad (4)$$

Number of Records	4,819,992,847
Number of Users	1,648,796
Number of Tracks	13,227,376
Date Range	January 2012 - February 2013

Table 1. Dataset of listening records.

This is similarly normalized across time periods to get new listen rates

$$\hat{R}_{ij} = \frac{\hat{E}_{ij}}{\sum_{l=1}^W \hat{E}_{il}}. \quad (5)$$

The intuition is to limit the effect of repeat plays among individual users. In estimating cultural preferences, there is likely more information gained when 100 users listen to a track once than when one user listens to a track 100 times.

2.2 Detection

For detection, [11] fit a multi-variate Gaussian with listen rates of only Christmas tracks with parameters $\theta_{Christmas}$ composed of mean, μ_{R_j} , and covariance matrix, Σ_{R_j} . Given a new track with listen rates for the same time periods, $\mathbf{x} = [R_1, R_2, \dots, R_W]$, the metric for detection was

$$P(\mathbf{x}|\theta_{Christmas}) = \prod_{j=1}^W \mathcal{N}(\mathbf{x}|\mu_{R_j}, \Sigma_{R_j}). \quad (6)$$

We propose another metric using the posterior probability from a direct application of Bayes' Rule in

$$P(\theta_c|\mathbf{x}) = \frac{P(\mathbf{x}|\theta_c) * P(c)}{P(\mathbf{x}|\theta_c) * P(c) + P(\mathbf{x}|\theta_n) * P(n)} \quad (7)$$

where c subscript represents Christmas and n subscript represents non-Christmas. This includes a model for non-Christmas tracks and prior probabilities $P(c)$ and $P(n)$, which represent the proportion of Christmas and non-Christmas tracks in matrix C , respectively. The priors in particular are important because of the small number of Christmas tracks in the dataset. $P(\mathbf{x}|\theta_n)$ is calculated from Eqn (6) where μ_{R_j} and Σ_{R_j} are calculated using all non-Christmas tracks.

2.3 Dataset

This study uses the same internal Gracenote dataset of online radio listening records in North America as [11]. Some basic information is shown in Table 1. Each record in the dataset represents one listen of a track by one user and provides User ID, Date, Time, and Track ID. Track metadata is also available such as track name, album name, and artist name.

Min. Listens	All Tracks	Christmas Tracks
1,500	338,406	4,732
500	767,116	10,647
200	1,397,032	18,170
100	2,087,863	26,134
10	5,906,307	68,582
1	10,207,335	118,515

Table 2. Track distribution at each threshold of minimum listens.

3. CHRISTMAS

3.1 Experiments

In these experiments, we compared the performance of the signals and prediction metrics in Section 2. As in [11], we generated a ground truth of Christmas tracks by searching for keyword “Christmas” in track names and album names. We defined the window radius, r_w , as the number of consecutive days before and after the target holiday, December 25, 2012, such that the window length $W = 2 * r_w + 1$. Since [11] showed an increase in performance with increasing popularity of tracks, we use the same thresholds of minimum listens in the dataset (1,500, 500, 200, 100, 10, 1) for direct comparison. Table 2 shows the distribution of tracks for each threshold.

For each listen rate in Section 2.1, a matrix was constructed from the dataset using tracks above the specified threshold, all users, and W days. We varied W by choosing r_w ranging 1 to 30 to capture the signal up to one month before and a month after December 25. The matrix was randomized and split into train (60%) and test (40%) sets on the first dimension. Two single component Gaussian Mixture Models, for Christmas and non-Christmas, were trained with the training set in a supervised manner with each training example a track and features the listen rates for each day in the signal window. Classification was performed with each metric in Section 2.2 on the test set and the area under the Receiver Operating Characteristic (AUROC) was calculated for evaluation.

3.2 Results

Figure 2 and Figure 3 show the AUROC against the window radius for the proposed listen rate, \hat{R} , and each prediction metric. Observing the difference in y-axis scale, the most notable difference between the figures is an increase in performance across all thresholds and signal lengths for the posterior probability. In particular, the lowest two thresholds have quite large increases of about 0.15 at each signal length.

Among tracks with the strongest listening signals, there is a small decay with increased window length. In contrast, the weakest listening signals show a large boost in performance with increased signal length. Similar plots for listen rates R are not shown because they track very closely and mostly just below the trends for \hat{R} . Lastly, Table 3 shows the maximum AUROC value for each threshold across sig-

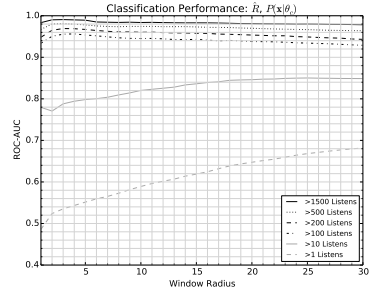


Figure 2. AUROC for each listen threshold for listen rate \hat{R} and prediction metric $P(\mathbf{x}|\theta_c)$

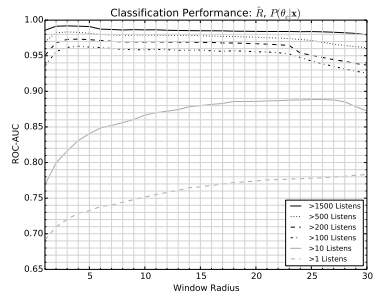


Figure 3. AUROC for each listen threshold for listen rate \hat{R} and prediction metric $P(\theta_c|\mathbf{x})$

nal lengths as a measure of overall performance. The proposed signal and prediction metric give the highest AUROC for the top four thresholds, and the signal from [11] with the proposed prediction metric have slightly higher AUROC for the bottom two thresholds.

3.3 Analysis

The posterior probability performs better than the likelihood because the inclusion of a non-Christmas model provides additional discriminative information. There is a lot of complexity in the non-Christmas tracks that is not modeled well by a single Gaussian with a mostly uniform distribution as shown in Figure 1. This suggests that incorporating models for other common signal shapes such as those of newly released tracks might further improve performance.

The signal length effects the performance in different ways. Tracks with the strongest listening signals perform best more with localized time window. We believe this is due primarily to higher variability of listening rates leading up to Christmas. The Christmas holiday is celebrated

Threshold	$P(x \theta_c)$		$P(\theta_c x)$	
	R	\hat{R}	R	\hat{R}
1,500	0.987	0.991	0.989	0.992
500	0.975	0.981	0.978	0.983
200	0.964	0.969	0.969	0.973
100	0.950	0.956	0.958	0.963
10	0.851	0.850	0.892	0.888
1	0.680	0.682	0.784	0.783

Table 3. Best AUROC for any signal window.

for many days before, and the signals during this time may be less stable than nearby December 25th. Tracks with the weakest listening signals perform best with a larger time window performs. This is likely because there is more information available with a longer signal, even if a small amount. Tracks with 50 plays in the dataset average only one play in ten days so capturing enough discriminatory information for detection requires a longer signal window.

The proposed signal of user counts, \hat{R} , has a smoothing effect over the signal of play counts, R , boosting performance. With tracks of stronger signals this appears to be more discriminating as shown in Table 3. But tracks with weaker signals, this seems to remove some useful information, which would explain why the play counts, R , perform slightly better at the two lowest thresholds. No single configuration appears to give optimal performance for this task.

4. HOLIDAY GENERALIZATION

We are interested in detecting track-temporal context associations for many days other than Christmas. Directly repeating the procedure in Section 3 for other holidays produced poor results on the dataset in Section 1. We believe this is because the ground truth generated from keywords is much less clean. Since many other holidays have a smaller music repertoire than Christmas, discriminative keywords like the holiday names generate too few tracks with strong listening signals for model training. And less discriminative keywords inadvertently include tracks not associated with the holiday, similarly compromising training.

Instead, the Christmas model in Section 2 can be reinterpreted as a holiday model with parameters θ_{holiday} composed of the same mean, μ_{R_j} , and covariance matrix, Σ_{R_j} as Eqn (6). Now a new track with listen rate signal of the same length, W , centered on a *different target holiday*, $\mathbf{x} = [R_1, R_2, \dots, R_W]$, can be detected with Eqn (6) or Eqn (7).

4.1 Experiments

In these experiments, we show the performance of detection on three other holidays. Since the dataset in Section 1 is from users in North America, we chose Halloween, Saint Patrick’s Day, and U.S. Independence Day as they are well-known holidays in North America and likely to have music associations. For the best results, we use only tracks with

Saint Patrick’s	U.S. Independence	Halloween
95%	73%	99%

Table 4. Average precision@50 for holiday track detection.

strong listening signals - above 1,500 total listens in the dataset - and the best performing listen rate and prediction metric from Section 3, \hat{R} and Eqn (7).

We constructed the training set feature matrix using $W = 15$, implying $r_w = 7$ and $\hat{R}_{i,8}$ is the listen rate for track i on December 25, 2012. Again, we trained two single component Gaussian Mixture Models, holiday and non-holiday, in a supervised manner. We constructed the test set feature matrix similarly with $W = 15$, meaning $\hat{R}_{i,8}$ is the listen rate for track i on the target holiday.

We calculated the probability of each track in the test set with Eqn (7) and ranked the tracks from highest probability to lowest for analysis. We chose precision@k to provide a general measure of relevance of tracks. We set k=50 with the assumption that there are at least 50 tracks truly associated with each holiday in order to better attribute errors in detection to the methodology. Three music content experts examined each list and labeled tracks as relevant to the holiday or not. We averaged the results to get a single value of precision@50 for each holiday.

4.2 Results

Table 4 shows the average precision@50 of holiday track detection as indicated by three music experts. Halloween and Saint Patrick’s Day had high values at 99% and 95%, respectively, and U.S. Independence Day was lower at 73%. The mean probability of the all tracks according to the holiday model was 99.9%. The distribution of incorrect tracks for U.S. Independence Day is skewed toward the bottom of the list.

The top 10 tracks for each holiday are shown in Section 4.2.1 - Section 4.2.3 to further characterize the results. All of these tracks had a probability of 1.0 according to the holiday model. The ordering for each track is track name, artist name.

4.2.1 Top 10 Saint Patrick’s Tracks

1. When Irish Eyes Are Smiling, Bing Crosby
2. Maloney Wants A Drink, The Clancy Brothers
3. Sally MacLennane, The Pogues
4. When Irish Eyes Are Smiling, The Irish Folk
5. Danny Boy, Irish Drinking Songs
6. Water Is Alright In Tay, The Clancy Brothers
7. Whiskey In The Jar, The Clancy Brothers
8. A Pair of Brown Eyes, The Pogues
9. The Black Velvet Band, Irish Drinking Songs
10. Grace, Jim McCann

4.2.2 Top 10 U.S. Independence Tracks

1. America, Barry White
2. Independence Day, Elliott Smith

3. Proud to be an American, Tiki
4. Stars And Stripes Forever, John Philip Sousa
5. Justice And Independence '85, John Mellencamp
6. 4th of July, X
7. Our Country (Rock Version), John Mellencamp
8. America the Beautiful, Blake Shelton and Miranda Lambert
9. This Is My Country, The Impressions
10. God Bless The U.S.A., Lee Greenwood

4.2.3 Top 10 Halloween Tracks

1. Purple People Eater, Halloween Hit Factory
2. "Dr. Who" Theme Song, Mannheim Steamroller
3. Graveyard Of The Living Dead, Halloween Sound Effects
4. Werewolves - Scary Halloween Sound Effects, Halloween Sound Effects
5. Dracula's Organ - Scary Halloween Sound Effects, Halloween Sound Effects
6. Creatures Of The Night (Original Mix), Mannheim Steamroller
7. This Is Halloween, The Countdown Kids
8. Hall of Screams - Scary Halloween Sound Effects, Halloween Sound Effects
9. Scary Halloween Haunted House, Sound Fx
10. Grimly Fiendish (Album Edit Version), The Damned

5. ANALYSIS - HOLIDAY

The high values for precision@50, particularly those of Halloween and Saint Patrick's, show that a model trained with user data around Christmas is effective in identifying daily music-temporal context associations. The lower precision@50 of U.S. Independence Day and the incorrect tracks being skewed towards the bottom of the list suggests that our assumption of at least 50 associated tracks for the holiday may be incorrect. Flaws in the methodology could also be the cause.

In particular, the assumption that Christmas listening signals have a distribution that matches those other holidays closely is likely flawed. Looking at Christmas signal in Figure 1 and the Saint Patrick's signal in Figure 4, they are similar but do not match exactly. The Christmas tracks have two peaks, December 24 and December 25, and the Saint Patrick's tracks have a single peak on March 17. With shorter signal lengths, this difference is pronounced and gives poor results for detecting other holiday tracks. This is why the experiments in Section 4.1 used $r_w = 7$, and not the optimal from Section 3, $r_w = 3$.

Among the incorrect U.S. Independence tracks, nearly one-third were from a single album by electro-punk band Frittenbude. This highlights other possible reasons for increased engagement such as marketing pushes. This album appears to have been released in the summer of 2012 and the synchronized rise and fall of the album's initial listening could be one explanation. In this case and other one-time events, album releases happen just once and could be separated from the more cyclical holiday listening with multiple years of data.

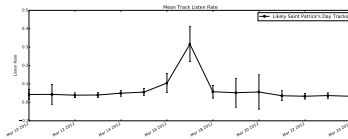


Figure 4. Mean listen rates R for top 100 predicted Saint Patrick's Day tracks from March 10, 2012 to March 24, 2012.

The effectiveness of general holiday detection implies one obvious commercial application: an automated, "always-on" seasonal radio station. With multiple years of data, the results likely could be improved and characterized by their change over time. Also, the addition of location data could highlight geographical differences for improved recommendations. For example, since our dataset is primarily North America, the tracks in Section 4.2.1 may be poor recommendations to users celebrating Saint Patrick's in Ireland or other parts of the world.

6. FUTURE WORK

The issues with matching the signal shapes of Christmas tracks to other holidays suggest room for improvement. Artificial templates or hand labeling a holiday ground truth could estimate the target distributions more accurately. Although labeling track-temporal context associations with user data has advantages over the other automated methods as outlined in Section 1, combining these methods could produce superior results. Lastly, applying this methodology at additional time resolutions (e.g hours, weeks, months) or exploring how these contexts interact with user data (e.g. age, geography, personality) could further enrich the user listening experience.

7. CONCLUSION

This study showed improvements to previous method for detecting Christmas tracks from user listening data and generalized the method to detect tracks for other holidays. The proposed improvements showed small increases of about 0.01 maximum AUROC for the most popular tracks but larger improvements of about 0.1 maximum AUROC for less popular tracks. Detection of Halloween, Saint Patrick's Day, and July 4th tracks was promising with precision@50 at 95%, 99%, and 73%, respectively.

8. REFERENCES

[1] Christmas comes early to The Echo Nest (The Echo Nest Blog) <http://blog.echonest.com/post/35845347430/christmas-comes-early-to-the-echo-nest>

[2] Top 50 Love Songs of All Time (Billboard)

- <http://www.billboard.com/articles/list/1538839/top-50-love-songs-of-all-time>
- [3] Last.fm (Last.fm) <http://www.last.fm/charts/toptags>
 - [4] P. Lamere. "Social Tagging and Music Information Retrieval." *Journal of New Music Research*, Vol. 37 No. 2 pp. 101-114, 2008.
 - [5] Reis, Ben Y., Marcello Pagano, and Kenneth D. Mandl. "Using temporal context to improve biosurveillance." *Proceedings of the National Academy of Sciences* 100.4 (2003): 1961-1965.
 - [6] Kulkarni, Anagha, et al. "Understanding temporal query dynamics." *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 2011.
 - [7] M. Shokouhi: Shokouhi, Milad. "Detecting seasonal queries by time-series analysis." *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*. ACM, 2011.
 - [8] Park, Chan Ho, and Minsuk Kahng. "Temporal dynamics in music listening behavior: A case study of on-line music service." *Computer and Information Science (ICIS), 2010 IEEE/ACIS 9th International Conference on*. IEEE, 2010.
 - [9] Carneiro, Mrio Joo Teixeira. "Towards the discovery of temporal patterns in music listening using Last.fm profiles." *Dissertation*, 2012.
 - [10] Hidasi, Balzs, and Domonkos Tikk. "Context-aware recommendations from implicit data via scalable tensor factorization." *arXiv preprint arXiv:1309.7611*, 2013.
 - [11] C. Summers, P. Popp. "Large Scale Discovery of Seasonal Music with User Data." *User Modeling, Adaptation and Personalization: 23rd International Conference, UMAP 2015, Extended Proceedings*. Springer, 2015.

IMPROVING MUSIC RECOMMENDATIONS WITH A WEIGHTED FACTORIZATION OF THE TAGGING ACTIVITY

Andreu Vall Marcin Skowron Peter Knees Markus Schedl
Department of Computational Perception, Johannes Kepler University, Linz, Austria
{andreu.vall, marcin.skowron, peter.knees, markus.schedl}@jku.at

ABSTRACT

Collaborative filtering systems for music recommendations are often based on implicit feedback derived from listening activity. Hybrid approaches further incorporate additional sources of information in order to improve the quality of the recommendations. In the context of a music streaming service, we present a hybrid model based on matrix factorization techniques that fuses the implicit feedback derived from the users' listening activity with the tags that users have given to musical items. In contrast to existing work, we introduce a novel approach to exploit tags by performing a weighted factorization of the tagging activity. We evaluate the model for the task of artist recommendation, using the expected percentile rank as metric, extended with confidence intervals to enable the comparison between models. Thus, our contribution is twofold: (1) we introduce a novel model that uses tags to improve music recommendations and (2) we extend the evaluation methodology to compare the performance of different recommender systems.

1. INTRODUCTION AND RELATED WORK

We provide the motivation of our work together with a review of the relevant related work, divided into three parts. First, we introduce the types of user feedback under consideration. Then, we present the family of models we use to build recommender systems. Finally, we review the evaluation methodology.

1.1 Explicit, Implicit and One-Class Feedback

The interactions between users and items provide a useful source of data to produce recommendations [16]. It is commonly accepted to distinguish between *explicit feedback* and *implicit feedback*, depending on whether the user actively provides feedback about an item or this is tracked from the user's interaction with the system [1]. Examples of explicit feedback are rating a movie, giving a "like" to a blog post, or tagging an artist, because the user actively

provides an opinion. In contrast, the listening histories of users in a music streaming service are an example of implicit feedback.

The standard approach to make use of implicit feedback is to count or aggregate all the interactions for each user-item pair [5, 7, 8], yielding a user-item-count table. In structure, this is identical to an explicit feedback user-item-rating table. We henceforth refer to such data structure as *user-item interactions matrix*, regardless of the type of feedback (implicit or explicit).


In some cases a user-item interaction can express both positive and negative opinions, in other cases it only reflects positive (or active) examples. Ratings in a 1 to 5 scale conventionally range from strongly disliking an item to strongly liking it. However, tracking whether a user visited or not a website, only provides a binary feedback describing action or inaction. Binary feedback is often referred to as *one-class feedback* [12, 15, 17], and examples of it can be found both in explicit and in implicit feedback. For example, a user-item interactions matrix (be it from explicit or implicit feedback) contains intrinsically a source of one-class feedback, revealing which user-item pairs were observed and which not.

Inaction must not be confused with a negative opinion, because a user may not have interacted with an item for a variety of reasons, not necessarily because of lack of interest. Social tags also exhibit this property, and treating this correctly will be a key point of the presented model.

1.2 Matrix Factorization for Collaborative Filtering

Collaborative filtering is a widely used recommendation method which aims at recommending the most relevant items to a user based on relations learned from previous interactions between users and items [16]. The factorization of the user-item interactions matrix into latent factors matrices is a well established technique to implement collaborative recommender systems, both for explicit feedback and implicit feedback datasets [7, 10, 15]. Compared to other methods, it has the advantage of uncovering latent data structures by solving an optimization problem, instead of using problem-specific and manually-designed features.

Specific collaborative systems for implicit feedback data based on matrix factorization techniques are presented in [7, 15]. The key technique is to use appropriate weights in the low-rank approximation of the user-item interactions matrix. More specifically, even if the weighting schemes are different, both [7, 15] assign higher confidence to the

 © Andreu Vall, Marcin Skowron, Peter Knees, Markus Schedl. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Andreu Vall, Marcin Skowron, Peter Knees, Markus Schedl. "Improving music recommendations with a weighted factorization of the tagging activity", 16th International Society for Music Information Retrieval Conference, 2015.

observed user-item pairs and lower (but still positive) confidence to the unobserved user-item pairs. This is important to handle the uncertainty derived from the one-class property described before. We will insist on this point later, because our improved treatment of the tagging activity will rest on the same principle.

1.3 Hybrid Recommender Systems

In collaborative filtering implementations based on matrix factorization techniques, hybrid models can be based on the simultaneous factorization of the user-item interactions matrix, together with other data for users and items [5, 13]. The motivation for that is that latent factors summarizing user and item properties should be reinforced, or better described, if other data sources related to the same users and items are involved in the optimization problem.

The tags that users assign to musical items –or other forms of textual data, like user profiles or genre annotations for the items– are an obvious example of potentially useful additional information. In this line, the research presented in [5] is a valid starting point, dealing with implicit feedback data and hybridized with user and item profiles, built on the basis of tf-idf weights calculated for each user, each item and each considered word in a dictionary.

Tagging information is an explicit source of feedback (because users actively provide it) that exhibits, at the same time, the one-class property described before; the tags assigned to musical items are only positive examples (even if the meaning of a tag is semantically negative). A particular tag may not have been applied to a musical item, but this does not imply that the tag is not suited to describe that musical item. This property of social tags is also referred to as *weak labeling* [18]. It is reasonable to assume though, that the more often a tag has been applied to a musical item, the more it should be trusted. Similarly, if a user applies a tag very often, it may be assumed that the tag is to some extent relevant for her to describe musical items. To address the uncertainty that arises from this wide range of possibilities, we propose to exploit the tagging activity with a weighted matrix factorization scheme similar to the one applied for collaborative filtering in implicit feedback datasets. Observed tags can be given higher confidence. Unobserved tags can be given lower confidence, but still positive, so that they are not ignored in the recommendation system.

1.4 Evaluation of Recommender Systems

The Netflix Prize [3] has motivated an important progress in the domain of collaborative filtering, but probably due to the specific approach considered in the challenge, research has centered on attaining maximum levels of accuracy in the prediction of ratings. However, improvements in predictive accuracy do not always translate to improved user satisfaction [14].

To make the evaluation task more similar to a real use case (although still in an off-line experiment), [9] evaluates different recommender systems on the basis of issued ranked lists of recommendations. A recommender able to

rank first the relevant items should be considered better than a recommender that is not able to do so. An extension of this evaluation methodology to deal with implicit feedback datasets is proposed in [7] and applied in [8, 12]. It consists in a central tendency measure, called *expected percentile rank*, assessing how good is the recommender at identifying relevant items.

The expected percentile rank is a valid metric to measure the average behavior of a single recommender system, but in order to compare the performance of different recommender systems, considering only mean values can be inaccurate. We propose to use bootstrapping techniques to examine the distribution of the expected percentile rank and test for significant differences between models.

2. METHODOLOGY

This work is framed in the context of music streaming services in which users interact with musical items, mainly listening to music, but also through the free input of text describing them. We focus on the task of artist recommendations. The listening data is aggregated at the artist level, obtaining a user-artist-count matrix of implicit feedback. The tagging activity yields a user-artist-tag matrix of one-class feedback, processed to obtain: a user-tag-count matrix, describing how many times a user applied a tag, and an artist-tag-count matrix, describing how many times a tag was applied to an artist. The proposed model is actually flexible regarding the tagging activity data. In our experiments, we successfully use a collection of top used tags (not a complete list of all the used tags) together with weights describing the tag relevance (instead of actual counts).

2.1 Recommender System Models

We compare three recommender systems. The first is a standard collaborative filtering model for implicit feedback data. The second is a hybrid model incorporating textual data, that we modify for the specific task of using tags. Finally, we introduce a novel model, able to improve the quality of the recommendations through a weighted factorization of the tagging activity.

2.1.1 Implicit Feedback Matrix Factorization (MF)

We use the approach described in [7] to perform collaborative filtering on implicit feedback data. It consists in a weighted low-rank approximation of the user-artist-count matrix, adjusting the confidence of each user-artist pair as a function of the count. Given a system with N users and M artists, the counts for each user-artist pair are tabulated in a matrix $R \in \mathbb{N}^{N \times M}$, where users are stored row-wise and artists column-wise. A binary matrix \tilde{R} is defined, such that for each user u and each artist a

$$\tilde{R}_{ua} = \begin{cases} 1 & \text{if } R_{ua} > 0 \\ 0 & \text{if } R_{ua} = 0 \end{cases}, \quad (1)$$

and the following weight function is defined as

$$w(\eta, x) = 1 + \eta \log(1 + x). \quad (2)$$

Other weight functions can be defined and may better suit each specific problem and distribution of the data. We choose a logarithmic relation (instead of the also common linear relation used in [5, 7, 8]) to counteract the long-tail distribution of the data, where a majority of users have a small percentage of the total observed interactions. However, the detailed optimization of this function is not within the scope of this work.

Finally, the matrix factorization consists in finding two D -rank matrices $P \in \mathbb{R}^{N \times D}$ and $Q \in \mathbb{R}^{M \times D}$ (rows are latent features for users and artists respectively) minimizing the following cost function:

$$J_{MF}(P, Q) = \sum_{ua \in R} w(\alpha, R_{ua}) \left(\tilde{R}_{ua} - P_u Q_a^T \right)^2 + \lambda \left(\|P\|_F^2 + \|Q\|_F^2 \right). \quad (3)$$

Matrix \tilde{R} is reconstructed using P and Q . \tilde{R}_{ua} is the entry of \tilde{R} corresponding to user u and artist a . P_u is the row of P corresponding to user u , and Q_a is the row of Q corresponding to artist a . The squared reconstruction error is weighted using a function of the actual counts in R_{ua} according to equation (2) and it is summed over all the user-artist pairs.¹ The parameter α contributes to the weight function and is determined by grid search. A regularization term involving the Frobenius norm of P and Q is added to prevent the model from over-fitting. The regularization parameter λ is also determined by grid search.

2.1.2 Implicit Feedback Matrix Factorization with Tagging Activity (TMF)

Equation (3) is extended in [5] to incorporate textual information. We present a modification of this model to specifically deal with tags. Given a system where T tags have been used, the counts for each user-tag pair are stored in a matrix $T^U \in \mathbb{N}^{N \times T}$, where rows correspond to users and columns correspond to tags. The counts for each artist-tag pair are stored in a matrix $T^A \in \mathbb{N}^{M \times T}$, where rows correspond to artists and columns correspond to tags. The modified model factorizes together \tilde{R} , T^U and T^A into three D -rank matrices $P \in \mathbb{R}^{N \times D}$, $Q \in \mathbb{R}^{M \times D}$, $X \in \mathbb{R}^{T \times D}$ (rows are latent features for users, artists and tags respectively) minimizing the following cost function:

$$J_{TMF}(P, Q, X) = \sum_{ua \in R} w(\alpha, R_{ua}) \left(\tilde{R}_{ua} - P_u Q_a^T \right)^2 + \mu_1 \sum_{ut \in T^U} (T_{ut}^U - P_u X_t^T)^2 + \mu_2 \sum_{at \in T^A} (T_{at}^A - Q_a X_t^T)^2 + \lambda \left(\|P\|_F^2 + \|Q\|_F^2 + \|X\|_F^2 \right). \quad (4)$$

The first term is identical as in (3). The second and third terms account for the contribution of tags. X_t is the row

¹ As described in [7], this includes the zero entries of R as well.

of X corresponding to tag t . Matrices T^U and T^A are reconstructed using P , Q and X , and the squared reconstruction errors are summed over all user-tag pairs and artist-tag pairs. The parameters μ_1, μ_2 account for the contribution of each term to the cost function, and are determined by grid search. The regularization term is analogous as in (3).

This formulation modifies the one described in [5], in that it factorizes T^U and T^A using a single shared tags' factor matrix X , instead of two dedicated factor matrices. The tagging activity consists of user-artist-tag observations. Even if we use separated user-tag-count and artist-tag-count matrices as inputs for the model, the tags must be factorized in the same space of latent features.

This model factorizes the user-tag and artist-tag raw counts. If, for example, an artist-tag pair has never been observed, the model will try to fit a value of 0 counts for it. This seems an unsuited model, because we know that a tag that has not been applied may still be relevant.

2.1.3 Implicit Feedback Matrix Factorization with Weighted Tagging Activity (WTMF)

We introduce a novel approach to improve the hybridization with tagging activity, by using a weighted factorization scheme similar to the one used for implicit feedback data. The observed user-tag and artist-tag pairs are given high confidence and therefore have a higher contribution to the cost function. The unobserved user-tag and artist-tag pairs are given low confidence. They become less relevant in the cost function, and at the same time the model has more freedom to fit them. As the results in Section 3.3 demonstrate, this is a better approach to model the weak labeling property of social tags.

We define binary matrices \tilde{T}^U and \tilde{T}^A , such that for each user u , each artist a and each tag t

$$\tilde{T}_{ut}^U = \begin{cases} 1 & \text{if } T_{ut}^U > 0 \\ 0 & \text{if } T_{ut}^U = 0 \end{cases} \quad (5)$$

$$\tilde{T}_{at}^A = \begin{cases} 1 & \text{if } T_{at}^A > 0 \\ 0 & \text{if } T_{at}^A = 0 \end{cases}.$$

We factorize together \tilde{R} , \tilde{T}^U and \tilde{T}^A into three D -rank matrices $P \in \mathbb{R}^{N \times D}$, $Q \in \mathbb{R}^{M \times D}$ and $X \in \mathbb{R}^{T \times D}$ (rows are latent features for users, artists and tags respectively) minimizing the following cost function:

$$J_{WTMF}(P, Q, X) = \sum_{ua \in R} w(\alpha, R_{ua}) \left(\tilde{R}_{ua} - P_u Q_a^T \right)^2 + \mu_1 \sum_{ut \in T^U} w(\beta, T_{ut}^U) \left(\tilde{T}_{ut}^U - P_u X_t^T \right)^2 + \mu_2 \sum_{at \in T^A} w(\gamma, T_{at}^A) \left(\tilde{T}_{at}^A - Q_a X_t^T \right)^2 + \lambda \left(\|P\|_F^2 + \|Q\|_F^2 + \|X\|_F^2 \right). \quad (6)$$

The equation is similar to (4), but now all the terms involve a weighted factorization. Note that the second and

third terms have specific weight coefficients β and γ , determined by grid search.

2.2 Parameter Estimation

Alternating Least Squares (ALS) is usually the preferred method to minimize the objective functions of models based on matrix factorization [2, 5–8, 15, 19]. ALS is an iterative method, where subsequently all but one of the factor matrices are kept fixed. This results in quadratic functions that approximate the original one. At each step, the cost value is expected to move closer to a local minimum and the process is repeated until convergence. Since the approximated functions are quadratic, the exact solution for the factors can be computed in closed form.

For each of the presented models, we provide the exact solution for the factors of each user u stored in P_u , each artist a stored in Q_a and each tag t stored in X_t . We introduce some additional notation. R_{r_u} , R_{c_a} , $T_{r_c}^U$, $T_{c_t}^U$, $T_{r_a}^A$, $T_{c_t}^A$ refer to the u^{th} , a^{th} , t^{th} row or column (r, c) of the corresponding matrix (R, T^U, T^A).² We also need to define the following matrices:

- $W_R^u \in \mathbb{R}^{M \times M}$ is a diagonal matrix with the weights computed for the u^{th} row of R in the diagonal
- $W_R^a \in \mathbb{R}^{N \times N}$ is a diagonal matrix with the weights computed for the a^{th} column of R in the diagonal
- $W_{T^U}^u \in \mathbb{R}^{T \times T}$ is a diagonal matrix with the weights computed for the u^{th} row of T^U in the diagonal
- $W_{T^U}^{c_t} \in \mathbb{R}^{N \times N}$ is a diagonal matrix with the weights computed for the t^{th} column of T^U in the diagonal
- $W_{T^A}^{r_a} \in \mathbb{R}^{T \times T}$ is a diagonal matrix with the weights computed for the a^{th} row of T^A in the diagonal
- $W_{T^A}^{c_t} \in \mathbb{R}^{M \times M}$ is a diagonal matrix with the weights computed for the t^{th} column of T^A in the diagonal

2.2.1 Solution for J_{MF}

For each user u and artist a , the latent factors are given by

$$\begin{cases} P_u = (Q^T W_R^u Q + \lambda I)^{-1} (Q^T W_R^u R_{r_u}^T) \\ Q_a = (P^T W_R^a P + \lambda I)^{-1} (P^T W_R^a R_{c_a}^T) \end{cases} \quad (7)$$

2.2.2 Solution for J_{TMF}

For each user u , artist a and tag t , the latent factors are given by

$$\begin{cases} P_u = (Q^T W_R^u Q + \mu_1 X^T X + \lambda I)^{-1} \\ \quad \quad \quad (Q^T W_R^u R_{r_u}^T + \mu_1 X^T T_{r_a}^{UT}) \\ Q_a = (P^T W_R^a P + \mu_2 X^T X + \lambda I)^{-1} \\ \quad \quad \quad (P^T W_R^a R_{c_a}^T + \mu_2 X^T T_{r_a}^{AT}) \\ X_t = (\mu_1 P^T P + \mu_2 Q^T Q + \lambda)^{-1} \\ \quad \quad \quad (\mu_1 P^T T_{c_t}^{UT} + \mu_2 Q^T T_{c_t}^{AT}) \end{cases} \quad (8)$$

² T^U and T^A may be further transposed, reading T^{UT} and T^{AT} .

2.2.3 Solution for J_{WTFM}

For each user u , artist a and tag t , the latent factors are given by

$$\begin{cases} P_u = (Q^T W_R^u Q + \mu_1 X^T W_{T^U}^{r_u} X + \lambda I)^{-1} \\ \quad \quad \quad (Q^T W_R^u R_{r_u}^T + \mu_1 X^T W_{T^U}^{r_u} T_{r_a}^{UT}) \\ Q_a = (P^T W_R^a P + \mu_2 X^T W_{T^A}^{r_a} X + \lambda I)^{-1} \\ \quad \quad \quad (P^T W_R^a R_{c_a}^T + \mu_2 X^T W_{T^A}^{r_a} T_{r_a}^{AT}) \\ X_t = (\mu_1 P^T W_{T^U}^{c_t} P + \mu_2 Q^T W_{T^A}^{c_t} Q + \lambda)^{-1} \\ \quad \quad \quad (\mu_1 P^T W_{T^U}^{c_t} T_{c_t}^{UT} + \mu_2 Q^T W_{T^A}^{c_t} T_{c_t}^{AT}) \end{cases} \quad (9)$$

2.3 Producing Recommendations

The technique employed to produce recommendations is the same for all the models. Once the factor matrices P , Q and X are learned, the user-artist preferences are predicted as $Z = PQ^T$. Note that the tags' factor matrix X is not directly involved in the prediction, although it contributed to a better estimation of P and Q . The new matrix Z is expected to be a reconstruction of \hat{R} for the observed user-artist pairs. For unobserved entries, Z is expected to reveal potential preferences on the basis of the learned user and artist factors. The closer a predicted user-artist preference is to 1, the more confidence we have that it corresponds to an interesting artist for the user. For each user u , a recommendation list is prepared showing the artists with higher predicted preference values in Z_u .

3. EXPERIMENTAL STUDY

3.1 Dataset

We compare the different models on a dataset of Last.fm listening histories, top tags used by users and top tags applied to artists, collected through the Last.fm API.³ The combination of the standard Taste Profile Subset⁴ with the Last.fm tags dataset⁵ would seem a preferable choice, but the absence of users' tagging activity makes it unsuitable.

The dataset is built as a stable subset of a running crawl of Last.fm listening events. The original crawl includes only users with non-empty country information, non-empty gender information and a value in the age field between 10 and 80 years, although such filtering is actually not needed. There is no constraint on the minimum or maximum number of artists a user has listened to. However, we only include users such that at least 95% of their listened artists have a valid MusicBrainz⁶ identifier, which is required to accurately crawl the artists' tagging activity. This does not bias the dataset towards popular artists, because the MusicBrainz is an open and collaborative platform, including a wide variety of artists. The users' tagging activity is fetched with the Last.fm user names.

³ <http://www.last.fm/api>

⁴ <http://labrosa.cc.columbia.edu/millionsong/tasteprofile>

⁵ <http://labrosa.cc.columbia.edu/millionsong/lastfm>

⁶ <https://musicbrainz.org/>

# listened artists	# users
1 – 10	64
11 – 20	84
21 – 30	122
31 – 40	77
41 – 50	96
50 – 100	466
101 – 2,332	1,993
total	2,902

Table 1: Distribution of users per number of listened artists.

The dataset includes 21,852,559 listening events, relating to 2,902 users and 71,223 artists, yielding 687,833 non-zero user-artist-count entries. This corresponds to a matrix density of roughly 0.3%. Table 1 shows the distribution of users as a function of the number of artists they listened to.

The top tags for each user (if any) are provided together with a count variable describing how many times the user applied it. The top tags applied to an artist (if any) are provided together with a percentage relative to the most frequently applied tag [11]. Because the API functions only return the top tags, we only observe a partial set of the tagging activity. In addition, although the user is presented with previously used tags, she can always input free text. To overcome these limitations, we perform regularization and simplification operations to the tag strings, namely: replacements of genre abbreviations with their extended version, spelling corrections, removal of non-alphanumeric characters and mapping of different spelling variants to a unique tag string, resulting in a unified set of tokens. As this process is applied to the fetched tags, we are left with 630 unique tags for 600 users and 12,902 unique tags for 67,332 artists, among which 494 unique tags are identified as identical between the user and the artist list. Note that tags were found for most of the artists, but only for 20% of the users. Probably, only a small subset of active users use the tagging functionality.

The whole matrix of user-artist counts is used, although not all users or artists have related tagging activity. Tags are a complement whenever they are available.

3.2 Evaluation Methodology

The most reliable evaluation method for a recommender system is an actual large-scale on-line experiment, where real users interact with the system [16]. This requires a complex infrastructure which, unfortunately, is not within the scope of this work. Since we only have access to historical data, we can not measure how new recommendations would be perceived by the users. Furthermore, in contrast to explicit feedback applications, accuracy metrics for predicted ratings are not meaningful for implicit feedback. Therefore, we adopt the evaluation approach proposed in [9] and adapted in [7] to deal with implicit

feedback datasets in a recall-oriented setting and we additionally propose an extension to it.

The observed user-artist pairs are split into training and test sets to perform 5-fold cross validation, letting each user have approximately 80% of the listened artists in the training set and 20% in the test set. For each user-artist pair u, a assigned to the test set, a random list of artists (not including a) is drawn. The list is then ranked according to the preferences of user u , learned from the training set as explained in Section 2. Finally, a is inserted in the sorted list, and its percentile rank within the list is stored as $rank_{ua}$.⁷ If a is ranked among the top positions of the list, then its percentile rank is close to 0%. If it is ranked in last positions, then its percentile rank is close to 100%.

After this process is done over all the splits, $rank_{ua}$ is known for all the observed user-artist pairs in the dataset. Then, following [5, 7, 8], the *expected percentile rank* is defined as the weighted average of $rank_{ua}$ with weights given by the user-artist counts:

$$\overline{rank} = \frac{\sum_{ua \in R} R_{ua} rank_{ua}}{\sum_{ua \in R} R_{ua}}. \quad (10)$$

Correctly ranking a highly relevant artist is more important than correctly ranking a less relevant artist. Likewise, failing to recommend a highly relevant artist is worse than failing to recommend a less relevant one. Values of \overline{rank} close to 0% indicate that the recommender is able to correctly rank the relevant artists. Producing ranked lists uniformly at random results in an expected percentile rank of 50%. Ranking all the relevant items in the last position of the list results in an expected percentile rank of 100%.

We extend the evaluation methodology by building confidence intervals of \overline{rank} . This allows us to test for significant differences in the performance of models. We use basic bootstrap confidence intervals, based on the bootstrap distribution of the expected percentile rank (see [4]). For all the observed user-artist pairs in the dataset, random samples with replacement and with the same size as the dataset are drawn. For each sample of user-artist pairs, the expected percentile rank is computed. We repeat this step 1,000 times to obtain the bootstrap distribution of \overline{rank} . We then build 95% confidence intervals of \overline{rank} using the basic bootstrap scheme described in [4].

3.3 Model Comparison

The models are evaluated and compared for a varying number of latent factors D , and for a varying number of training iterations. On the one hand, we fix the number of iterations to 10 and evaluate the models with 5, 10, 20, 50, 100 latent factors. On the other hand, we fix the number of factors to 10 and evaluate the models for 5, 10, 20, 50, 100 training iterations. We choose 10 factors and 10 training

⁷ Lists of any length may be prepared, and the percentile rank provides a unified scale. We use lists of 100 artists in our experiments. According to our experience, longer lists do not yield significant differences.

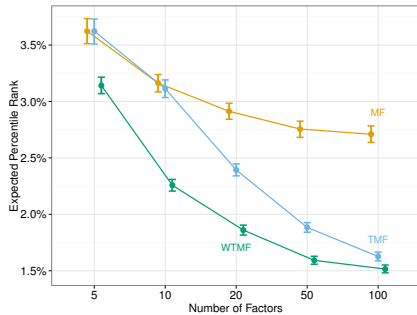


Figure 1: Model comparison for different number of latent factors. The dots correspond to \overline{rank} and the error bars display 95% basic bootstrap confidence intervals. The different models are dodged to avoid overlapping. The top and center lines correspond to the baseline models. The lowest corresponds to the presented model.

iterations as a basic setting, because they balance well performance and computational requirements.

For each model and each combination of factors and iterations, we tune the parameters α , β , γ , μ_1 , μ_2 and λ by grid search. We choose the set of values that provides lowest expected percentile rank, computed by 5-fold cross validation as described in Section 3.2. Figures 1 and 2 show the results for different number of factors and iterations respectively.

Note that all models, including the plain matrix factorization model, provide very good results, with values of expected percentile rank under 4%. This implies that, on average, the models are able to rank relevant artists among the top 4 positions of a list of 100 random artists.

The performance of TMF and WTMF improves significantly when more latent factors are used (see Figure 1). The presented model outperforms the baselines, although for 100 factors the difference between TMF and WTMF is small. We examine this case. We compute a 95% basic bootstrap confidence interval for the difference of \overline{rank} and it does not include 0. We conclude that the difference in performance is still significant. For lower number of factors the differences between the presented model and the baselines are remarkable. Good performance at inexpensive computational requirements is a crucial property, especially for large-scale implementations.

Increasing the number of training iterations results in smaller improvements (see Figure 2). Our model clearly outperforms the baselines in this set of experiments too, with a difference of nearly 1% in expected percentile rank. With the basic setting of 10 factors, TMF can not fully exploit the tagging activity and performs comparably to MF. For experiments with 20 or more training iterations they perform exactly as well, because the grid search process finds that discarding the tagging activity yields best results.

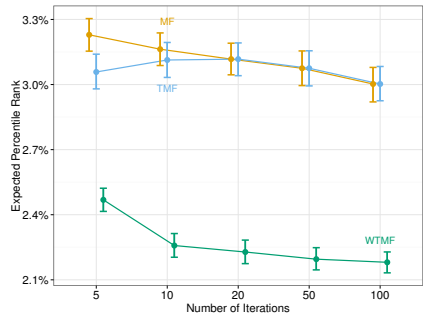


Figure 2: Model comparison for different number of training iterations. The dots correspond to \overline{rank} and the error bars display 95% basic bootstrap confidence intervals. The different models are dodged to avoid overlapping. The top lines correspond to the baseline models. The lowest corresponds to the presented model.

TMF performs slightly better than MF with 5 training iterations. The performance of TMF does not improve monotonically with more training iterations, although the model is not over-fitting. This is because after 5 iterations the cost function of TMF reaches a flat region close to a local minimum, resulting in small performance variations.

4. CONCLUSIONS AND FURTHER RESEARCH

In this paper we presented a novel model to incorporate tagging activity into implicit feedback recommender systems. Our approach proves to work better than previous hybrid models, based on experiments conducted with real data from Last.fm, a well-known music streaming service. We extended the common evaluation methodology computing basic bootstrap confidence intervals for the expected percentile rank. This allows us to test for significant differences in the performance of models.

As future work, we will evaluate the robustness of the presented model for different recommendation tasks. We are particularly interested in the task of song recommendations, but we will also experiment in fields other than music, like movies or websites. Another interesting question is the effect of the size and connectedness of the tagging data on the final quality of the recommendations. We will investigate how rich and linked together needs to be the tagging activity in order to enhance the recommendations. This could provide indications of when can the model be successfully utilized, or which kind of processing of the tag strings is required to make the tagging activity helpful.

5. ACKNOWLEDGMENTS

This research is supported by the Austrian Science Fund (FWF) under project no. P25655 and the EU FP7 through projects 601166 (PHENICX) and 610591 (GiantSteps).

6. REFERENCES

- [1] Gediminas Adomavicius and Alexander Tuzhilin. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 17(6):734–749, June 2005.
- [2] Robert M. Bell and Yehuda Koren. Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Proc. ICDM*, pages 43–52. IEEE, 2007.
- [3] James Bennett and Stan Lanning. The netflix prize. In *Proc. KDD Cup*, page 35, 2007.
- [4] Thomas J. DiCiccio and Bradley Efron. Bootstrap confidence intervals. *Statistical science*, pages 189–212, 1996.
- [5] Yi Fang and Luo Si. Matrix co-factorization for recommendation with rich side information and implicit feedback. In *Proc. HETREC*, pages 65–69. ACM, 2011.
- [6] K. Ruben Gabriel and S. Zamir. Lower Rank Approximation of Matrices by Least Squares with Any Choice of Weights. *Technometrics*, 21(4):489, November 1979.
- [7] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proc. ICDM*, pages 263–272. IEEE, 2008.
- [8] Christopher C. Johnson. Logistic Matrix Factorization for Implicit Feedback Data. 2014.
- [9] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proc. SIGKDD*, pages 426–434. ACM, 2008.
- [10] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.
- [11] M. Levy and M. Sandler. Music information retrieval using social tags and audio. *IEEE Transactions on Multimedia*, 11(3):383–395, April 2009.
- [12] Yanen Li, Jia Hu, ChengXiang Zhai, and Ye Chen. Improving one-class collaborative filtering by incorporating rich user information. In *Proc. CIKM*, pages 959–968. ACM, 2010.
- [13] Hao Ma, Tom Chao Zhou, Michael R. Lyu, and Irwin King. Improving recommender systems by incorporating social contextual information. *ACM Transactions on Information Systems*, 29(2):1–23, April 2011.
- [14] Sean M. McNee, John Riedl, and Joseph A. Konstan. Being accurate is not enough: How accuracy metrics have hurt recommender systems. In *Proc. CHI'06 Extended Abstracts*, pages 1097–1101. ACM, 2006.
- [15] Rong Pan, Yunhong Zhou, Bin Cao, Nathan Nan Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *Proc. ICDM*, pages 502–511. IEEE, 2008.
- [16] Francesco Ricci, Lior Rokach, Bracha Shapira, and Paul B Kantor, editors. *Recommender systems handbook*. Springer, 2011.
- [17] Vikas Sindhwani, Serhat S. Bucak, Jianying Hu, and Aleksandra Mojsilovic. One-class matrix completion with low-density factorizations. In *Proc. ICDM*, pages 1055–1060. IEEE, 2010.
- [18] Douglas Turnbull, Luke Barrington, and Gert Lanckriet. Five approaches to collecting tags for music. In *Proc. ISMIR*, volume 8, pages 225–230, 2008.
- [19] Yunhong Zhou, Dennis Wilkinson, Robert Schreiber, and Rong Pan. Large-scale parallel collaborative filtering for the netflix prize. In *Algorithmic Aspects in Information and Management*, pages 337–348. Springer, 2008.

An Efficient State-Space Model for Joint Tempo and Meter Tracking

Florian Krebs, Sebastian Böck, and Gerhard Widmer

Department of Computational Perception
Johannes Kepler University, Linz, Austria

florian.krebs@jku.at

ABSTRACT

Dynamic Bayesian networks (e.g., Hidden Markov Models) are popular frameworks for meter tracking in music because they are able to incorporate prior knowledge about the dynamics of rhythmic parameters (tempo, meter, rhythmic patterns, etc.). One popular example is the *bar pointer model*, which enables joint inference of these rhythmic parameters from a piece of music. While this allows the mutual dependencies between these parameters to be exploited, it also increases the computational complexity of the models. In this paper, we propose a new state-space discretisation and tempo transition model for this class of models that can act as a drop-in replacement and not only increases the beat and downbeat tracking accuracy, but also reduces time and memory complexity drastically. We incorporate the new model into two state-of-the-art beat and meter tracking systems, and demonstrate its superiority to the original models on six datasets.

1. INTRODUCTION

Building machines that mimic the human understanding of music is vital for a variety of tasks, such as organising and managing today’s huge music collections. In this context, automatic inference of metrical structure from a musical audio signal plays an important role. Generally, the metrical structure of music builds upon a hierarchy of approximately regular pulses with different frequencies. In the centre of this hierarchy is the *beat*, a pulse to which humans choose to tap their feet. These beats are again grouped into bars, with the *downbeat* denoting the first beat of each bar.

Several approaches have been proposed for tackling the problem of automatic inference of meter (or subcomponents such as beats and downbeats) from an audio signal, with approaches based on machine learning currently being the most successful [1, 5, 12, 13, 22]. All of these approaches incorporate probabilistic models, but with different model structures: the systems introduced in [5, 13, 22] decouple tempo detection from the detection of the

beat/downbeat phase, which has the advantage of reducing the search space of the algorithms but can be problematic if the tempo detection is erroneous. Others [1, 12] model tempo and beat/downbeat jointly, taking into account their mutual dependency, which leads to increased model complexity.

One popular model that jointly models tempo and bar position is the *bar pointer model*, first proposed in [20]. In addition to tempo and bar position, the model also integrates various rhythmic pattern states. It has been extended by various authors: in [12, 14] the benefit of using rhythmic pattern states to analyse rhythmically diverse music was demonstrated, in [18] a simplification for models with multiple rhythmic pattern states was proposed, in [17] the label of an acoustic event was additionally modelled in order to enable a drum robot to distinguish different instruments, and in [6] it was applied to a drum transcription task. These algorithms share the problem of a high space and time complexity because of the huge state-space in which they perform inference. In order to make inference tractable, the state-space is usually divided into discrete cells, with either fixed [1, 6, 12, 14, 17, 20] or dynamic [15, 18, 21] locations in the state-space. While the former approach can be formulated as a hidden Markov model (HMM), which performs best but is prohibitively complex, the latter uses particle filtering (PF), which is fast but performs slightly worse in sub-tasks such as downbeat tracking [15].

In this paper, we propose a modified bar pointer model which not only increases beat and downbeat tracking accuracy, but also reduces drastically time and memory complexity. In particular, we propose (a) a new (fixed grid) discretisation of the joint tempo and beat/bar state-space and (b) a new tempo transition model. We incorporated the new model into two state-of-the-art beat and meter tracking systems, and demonstrate its superiority on six datasets.

2. METHOD

In this section, we describe how we tackle the problem of metrical structure analysis using a probabilistic state-space model. In these models, a sequence of *hidden variables*, which in our case represent the meter of an audio piece, is inferred from a sequence of *observed variables*, which are extracted from the audio signal. For ease of presentation, we now consider a state-space of two hidden variables, the position within a bar and the tempo. Including additional hidden variables, e.g., a rhythmical pattern

 © Florian Krebs, Sebastian Böck, and Gerhard Widmer. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Florian Krebs, Sebastian Böck, and Gerhard Widmer. “An Efficient State-Space Model for Joint Tempo and Meter Tracking”, 16th International Society for Music Information Retrieval Conference, 2015.

state [12, 14, 18, 20, 21] or an acoustic event label [6, 17] is straightforward. In the following, we describe the original bar pointer model [20], its shortcomings, and the proposed improvements.

2.1 The original bar pointer model

The bar pointer model [20] describes the dynamics of a hypothetical pointer which moves through the space of the hidden variables throughout a piece of music. At each time frame k , we refer to the (hidden) state of the bar pointer as $\mathbf{x}_k = \{\Phi_k, \dot{\Phi}_k\}$, with $\Phi_k \in \{1, 2, \dots, M\}$ denoting the position within a bar, and $\dot{\Phi}_k \in \{\dot{\Phi}_{min}, \dot{\Phi}_{min} + 1, \dots, \dot{\Phi}_{max}\}$ the tempo in bar positions per time frame. M is the total number of discrete positions per bar, $N = \dot{\Phi}_{max} - \dot{\Phi}_{min} + 1$ is the total number of distinct tempi, $\dot{\Phi}_{min}$ and $\dot{\Phi}_{max}$ are respectively the lowest and the highest tempo. See Fig. 1a for an illustration of such a state space. Finally, we denote the observation features as \mathbf{y}_k .

Overall, we want to compute the most likely hidden state sequence $\mathbf{x}_{1:K}^* = \{\mathbf{x}_1^*, \mathbf{x}_2^*, \dots, \mathbf{x}_K^*\}$ given a sequence of observations $\{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_K\}$ for each audio piece as

$$\mathbf{x}_{1:K}^* = \arg \max_{\mathbf{x}_{1:K}} P(\mathbf{x}_{1:K} | \mathbf{y}_{1:K}). \quad (1)$$

with

$$P(\mathbf{y}_{1:K} | \mathbf{x}_{1:K}) \propto P(\mathbf{x}_1) \prod_{k=2}^K P(\mathbf{x}_k | \mathbf{x}_{k-1}) P(\mathbf{y}_k | \mathbf{x}_k). \quad (2)$$

Here, $P(\mathbf{x}_1)$ is the *initial state distribution*, $P(\mathbf{x}_k | \mathbf{x}_{k-1})$ is the *transition model*, and $P(\mathbf{y}_k | \mathbf{x}_k)$ is the *observation model*, which we further describe in the bottom of this section. Eq. 1 can be solved using the well-known Viterbi algorithm [16]. Finally, the set of downbeat frames \mathcal{D} can be extracted from the sequence of bar positions as

$$\mathcal{D} = \{k : \Phi_k^* = 1\}, \quad (3)$$

and the set of beat frames can be obtained analogously by selecting the time frames which correspond to a bar position that matches a beat position.

2.1.1 Initial distribution

Here, any prior knowledge (e.g., about tempo distributions) can be incorporated into the model. Like most systems, we use a uniform distribution in this work.

2.1.2 Transition model

The transition model $P(\mathbf{x}_k | \mathbf{x}_{k-1})$ can be further decomposed into a distribution for each of the two hidden variables Φ_k , and $\dot{\Phi}_k$ by:

$$P(\mathbf{x}_k | \mathbf{x}_{k-1}) = P(\Phi_k | \Phi_{k-1}, \dot{\Phi}_{k-1}) \cdot P(\dot{\Phi}_k | \dot{\Phi}_{k-1}). \quad (4)$$

The first factor is

$$P(\Phi_k | \Phi_{k-1}, \dot{\Phi}_{k-1}) = \mathbb{1}_x, \quad (5)$$

where $\mathbb{1}_x$ is an indicator function that equals one if $\Phi_k = (\Phi_{k-1} + \dot{\Phi}_{k-1} - 1) \bmod M + 1$, and zero otherwise. The modulo operator makes the bar position cyclic (the last, light grey column in Fig. 1a is identical to the first column).

The second factor $P(\dot{\Phi}_k | \dot{\Phi}_{k-1})$ is implemented by

$$P(\dot{\Phi}_k | \dot{\Phi}_{k-1}) = \begin{cases} 1 - p_\Phi, & \dot{\Phi}_k = \dot{\Phi}_{k-1}; \\ \frac{p_\Phi}{2}, & \dot{\Phi}_k = \dot{\Phi}_{k-1} + 1; \\ \frac{p_\Phi}{2}, & \dot{\Phi}_k = \dot{\Phi}_{k-1} - 1, \end{cases} \quad (6)$$

otherwise $P(\dot{\Phi}_k | \dot{\Phi}_{k-1}) = 0$.

p_Φ is the probability of a tempo change. From Eq. 6 it can be seen that the pointer can perform three tempo transitions from each state (indicated by arrows in Fig. 1a).

2.1.3 Observation model

In this paper, we use two different observation models: The first one uses *recurrent neural networks* to derive a probability of a frame being a beat or not [1]. The second one models the observation probabilities with Gaussian mixture models from a two-dimensional onset feature [12, 14]. As the focus of this paper lies on the state discretisation and the tempo transition model, the reader is referred to [1, 12, 14] for further details.

2.2 Shortcomings of the original model

Previous implementations of the bar pointer model [1, 2, 6, 12, 14, 17] followed [20] in dividing the tempo-position state space into equidistant points, with each point aligned to an integer-valued bar position and tempo (see Fig. 1a). This discretisation has a number of drawbacks, which are further explained in the following.

2.2.1 Time resolution

As shown in Fig. 1a, the number of position grid points per bar is constant across the tempi. This means that the grid of a bar played at a low tempo has a lower time resolution than of a bar played at high tempo, because both are divided into the same number of cells. In contrast, there are more observations available for a bar at a low tempo than for a bar at a high tempo, since the observations are extracted at a constant frame rate. This causes a mismatch between the time resolution of the feature extraction and the time resolution of the discretised bar position.

2.2.2 Tempo resolution

As shown in Fig. 1a, the distance between two adjacent tempo grid points is constant across the grid. This is inconsistent with tempo sensitivity experiments on humans, which have shown that the human ability to notice tempo changes is proportional to the tempo, with the JND (just noticeable difference) being around 2-5% of the inter beat interval [4]. Therefore, in order to get a sufficiently high tempo resolution at lower tempi, a huge number of tempo states has to be chosen.

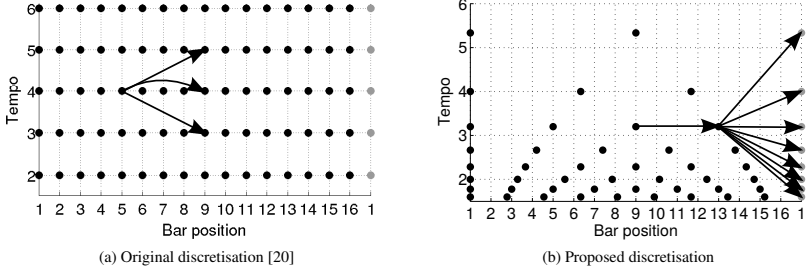


Figure 1: Toy example with $M = 16$ and $N = 6$: Each dot corresponds to a (hidden) state in the tempo-bar-position state-space. The arrows indicate examples of possible state transitions.

2.2.3 Tempo stability

As the tempo model (see Eq. 6) forms a first-order Markov chain, the current tempo state is independent of all tempo states given the past tempo state. This means that the tempo model is not able to reflect any long term dependencies between tempo states, which may result in unstable tempo trajectories.

2.3 Proposed model

This section introduces a solution to the problems described above. To simplify notation we assume a bar has four beats. Extending to other time signatures [20] or modelling beats instead of bars [1] is straightforward.

2.3.1 Time resolution

We propose making the number of discrete bar positions M dependent on the tempo by using exactly one bar position state per audio frame (and thus per observation feature value). The number of observations per bar (four beats at a tempo T in beats per minute (BPM) is

$$M(T) = \text{round}\left(\frac{4 \times 60}{T * \Delta}\right) \quad (7)$$

with Δ being the audio frame length. Using Eq. 7, we compute the number of bar positions of the tempo limits $M(T_{min})$ and $M(T_{max})$.

2.3.2 Tempo resolution

We can now either model all N_{max} tempi that correspond to integer valued bar positions in the interval $[M(T_{max}), M(T_{min})]$, with

$$N_{max} = M(T_{min}) - M(T_{max}) + 1, \quad (8)$$

or select only a subset of N tempo states. In Section 3, we evaluate the performance of the transition model for various numbers of tempo states. For $N < N_{max}$, we choose the tempo states by distributing N states logarithmically across the range of beat intervals, trying to mimic the JNDs of the human auditory system [4].

2.3.3 Tempo stability

To increase the stability of the tempo trajectories we only allow transitions at beat positions within a bar. This is illustrated in Fig. 1b with the arrows showing examples of possible state transitions. In contrast to the original model which allows three tempo transitions at every time step, we allow transitions to each tempo, but only at beat times. The new tempo transition model then becomes:

If $\Phi_k \in \mathcal{B}$,

$$P(\dot{\Phi}_k | \dot{\Phi}_{k-1}) = f(\dot{\Phi}_k, \dot{\Phi}_{k-1})$$

else

$$P(\dot{\Phi}_k | \dot{\Phi}_{k-1}) = \begin{cases} 1, & \dot{\Phi}_k = \dot{\Phi}_{k-1}; \\ 0, & \text{otherwise} \end{cases} \quad (9)$$

\mathcal{B} is the set of bar positions that corresponds to beats, and $f(\cdot)$ is a function that models the tempo change probabilities. We experimented with various functions (Gaussian, Log-Gaussian, Gaussian mixtures), but found this exponential distribution to be performing best:

$$f(\dot{\Phi}_k, \dot{\Phi}_{k-1}) = \exp(-\lambda \times |\frac{\dot{\Phi}_k}{\dot{\Phi}_{k-1}} - 1|) \quad (10)$$

where the rate parameter $\lambda \in \mathbb{Z}_{\geq 0}$ determines the steepness of the distribution. A value of $\lambda = 0$ means that transitions to all tempi are equally probable. In practice, for music with roughly constant tempo, we set $\lambda \in [1, 300]$. Fig. 2 shows the tempo transition probabilities for various values of λ .

2.4 Complexity of the inference algorithm

In this section, we investigate time and memory complexity of the bar pointer model, considering only the complexity of the (Viterbi) inference and ignoring the contribution of computing the observation features and observation probabilities.

Both time and space complexity depend on the number of states of the model. The number of states, in turn, depends on the number of bar positions, the tempo ranges, the audio frame length, and the tempo resolution that we chose to model. Let us assume that we have a model with

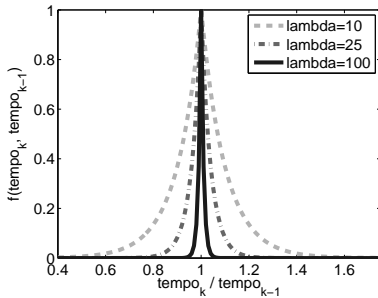


Figure 2: Tempo change probability density (Eq. 10) for various values of λ .

S hidden states, T possible state transitions per frame, and an audio excerpt with K frames. The memory requirement of the algorithm is then simply $S \times K$, as we have to store the best predecessor state for each of the S states for each time frame during Viterbi decoding. The time complexity, on the other hand, is $T \times K$, as we have to compute T transitions at each time step. In Table 1 we show the values of S and T of the models used in this paper.

3. EXPERIMENTAL SETUP

In this section, we evaluate the proposed model with real-world music data in two experiments¹. In the first experiment, we investigated the effect of the number of tempo states N and the rate parameter λ of the tempo transition function on the meter tracking performance on a training set. We evaluated only the beat tracking performance, as this is the most fundamental task that we wanted to solve. In the second experiment, we integrated the proposed model with the parameters determined in Experiment 1 into two state-of-the-art systems and compared the meter tracking performance in terms of accuracy and complexity with the original models. Below, we describe the datasets, the evaluation metrics, and the meter tracking models.

3.1 Datasets

In this work, we used seven test datasets, one for Experiment 1 and the remaining six for Experiment 2. For more details about each datasets, see the corresponding references:

Experimental dataset: This dataset is a subset of the *1360-songs* dataset [8] excluding the Hainsworth dataset, because it was used in Experiment 2. In total, it includes 1139 excerpts (total length 662 minutes).

Ballroom dataset [9]: A dataset of 698 30-second excerpts of ballroom dance music (total length 364 minutes).

¹ Additional information as well as the code to reproduce the results of this paper are available at <http://www.cp.jku.at/people/krebs/ismir2015/>

It was annotated with beat and downbeat times in [14].

Hainsworth dataset [10]: A dataset with 222 pieces (total length 199 minutes), covering a wide spectrum of genres.

SMC dataset [11]: A dataset with 217 pieces which are considered difficult for meter inference (total length 145 minutes). This set is also part of the MIREX evaluation.

Greek dataset [12]: 42 full songs of Cretan leaping dances in 2/4 meter (total length 140 minutes).

Turkish dataset [12]: 82 one-minute excerpts of Turkish Makam music (total length 82 minutes).

Indian dataset [19]: The same subset of 118 two-minute long pieces (total length 235 minutes) as used in [12].

3.2 Evaluation metrics

To assess the ability of an algorithm to infer metrical structure, we used five evaluation metrics - four for beat tracking and one for downbeat tracking.

F-Measure (F): computed from the number of true positives (correctly detected beats within a window of ± 70 ms around an annotation), the false positives, and the false negatives.

CMLt: quantifies the percentage of correctly tracked beats at the correct metrical level. In order to count a beat as correct, both previous and next beats have to match an annotation within a tolerance window of $\pm 17.5\%$ of the annotated beat interval.

AMLt: the same as CMLt, but the detected beats are also considered to be correct if they occur on the off-beat or at double or half of the ground-truth tempo.

Cemgil (Cem): places a Gaussian function with standard deviation of 40 ms around the annotations and computes the average likelihood of the corresponding beat closest to each annotation. In contrast to the other measures with hard decision boundaries (due to rectangular tolerance windows), this measure is also sensitive to small timing differences between annotated and detected beats.

Information Gain (D): measures the deviation of the beat error distribution from a uniform distribution by computing the Kullback-Leibler divergence.

Downbeat F-Measure (DB-F): is the same F-measure as used for beats, but considers only downbeats.

We implemented the evaluation metrics according to [3] with standard settings. To make them comparable with other work, we excluded the first five seconds in Experiment 2 when comparing with the model from [12] but did not exclude them when comparing with the results from [1].

3.3 Meter tracking models

To compare the proposed to the original model, we tested its performance with two state-of-the-art meter tracking systems:

RNN-BeatTracker [1]: This model uses a *recurrent neural network* to compute the probability of a frame being a beat. This probability is used as an observation probability for an HMM which jointly models tempo and the position

within a beat period. We used the same MultiModelBeatTracker model as described in [1]. The model uses a frame length of 10 ms. Only beats are detected with this model.

GMM-BarTracker [12, 14]: Gaussian Mixture Models (GMMs) are used to compute the observation probabilities for an HMM that jointly models tempo, position within a bar and a set of rhythmic bar-patterns. For Experiment 1, the GMMs were trained on the *Ballroom*, the *Beatles* [3], the *Hainsworth* and the *RWC_Popular* [7] datasets, using three rhythmic patterns that correspond to the time signatures 2/4, 3/4 and 4/4. Pieces with other time signatures were excluded. For Experiment 2, we used an updated² version of the model described in [12]. The model uses a frame length of 20 ms and integrates eight rhythmic pattern states, one for each of the rhythmic classes. It outputs beats and downbeats.

Note that the difference between *original* and *proposed* lies only in the definition of the hidden states and the transition model; both use the same observation model, initial distribution, and tempo ranges.

4. RESULTS AND DISCUSSION

4.1 Experiment 1

In this experiment, we evaluated the influence of two parameters of the proposed transition model on the meter tracking performance. These parameters are the width of the tempo change distribution parametrised by the rate λ (Section 2.3.3, Fig. 3) and the number of tempo states N (Section 2.3.1, Fig. 4). We chose to display the *Cemgil* accuracy in Figs. 3 and 4, because it is the only measure that makes a soft decision to count a beat as correct by using a Gaussian window and thus also takes into account small timing variations. Generally, the plots for the other measures were similar.

Fig. 3 shows the effect of the parameter λ on the *Cemgil* beat tracking accuracy for both the *RNN-BeatTracker* and the *GMM-BarTracker* on the *experimental* dataset, using the maximum number of tempo states N_{max} . The maximum *Cemgil* values were obtained with $\lambda = 125$, and $\lambda = 95$ respectively.

Using these settings for λ , we investigated the effect of the number of tempo states N on the beat tracking performance, which is shown in Fig. 4. As the two systems use a different audio frame rate, the maximum number of tempo states N_{max} is different too (see Section 2.3.1). Using a tempo range of [55, 215] BPM as in [1], the *RNN-BeatTracker* has at most $N_{max} = 82$ tempo states, while for the *GMM-BarTracker* $N_{max} = 41$. As can be seen from Fig. 4, the *Cemgil* accuracy converges at ≈ 75 tempo states for the *RNN-BeatTracker* and at ≈ 40 for the *GMM-BarTracker*. This finding suggests that the *BarTracker* might also benefit from a higher audio frame rate and therefore a higher number of tempo states. In addition, the number of tempo states is a suitable parameter to select a trade-off between speed and accuracy.

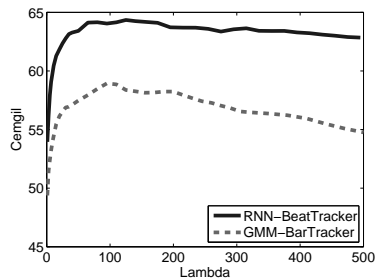


Figure 3: Effect of parameter λ on beat tracking *Cemgil* metric on the *experimental* dataset.

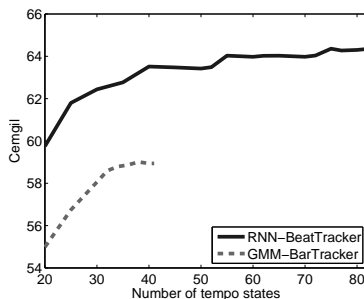


Figure 4: Effect of the number of tempo states on beat tracking *Cemgil* metric on the *experimental* dataset.

4.2 Experiment 2

In this experiment, we integrated the proposed model into two state-of-the-art meter tracking systems (Section 3.3) and compared them to the original models. The beat and downbeat accuracy scores of the original [1, 12] and proposed models, together with the number of states and transitions, are shown in Table 1. The proposed model used the parameters λ and N obtained in Experiment 1.

As can be seen, the proposed transition model outperforms the original model with respect to all performance metrics on all datasets (except *AMLt* (-0.2%) on the *Ballroom* dataset), with the added advantage of drastically reduced complexity. The *CMLt* metric in particular seems to benefit from the proposed model, with up to 20% relative improvement on the *Greek* dataset. Apparently, the restriction to change tempo only at beat times results in higher stability and therefore better performance in measures that are sensitive to continuity, such as *CMLt* and *AMLt*.

A comparison of the state-space sizes of the original and proposed models shows that the latter uses far fewer states and transitions. This is particularly apparent for the *GMM-*

² <http://www.cp.jku.at/people/krebs/ismir2014/>

	F	Cem	CMLt	AMLt	D	DB-F	States	Transitions
<i>RNN-BeatTracker</i>								
Ballroom								
Original [1] (20 tempo states)	0.910	0.845	0.830	0.924	3.469	-	11 520	33 280
Proposed (82 tempo states)	0.919	0.880+	0.854	0.922	3.552	-	5 617	8 343
Proposed (55 tempo states)	0.917	0.878	0.848	0.921	3.536	-	3 369	4 496
Hainsworth								
Original [1] (20 tempo states)	0.840	0.707	0.803	0.881	2.268	-	11 520	33 280
Proposed (82 tempo states)	0.851	0.730	0.805	0.885	2.337	-	5 617	8 343
Proposed (55 tempo states)	0.851	0.729	0.791	0.886	2.332	-	3 369	4 496
SMC								
Original [1] (20 tempo states)	0.529	0.415	0.428	0.567	1.460	-	11 520	33 280
Proposed (82 tempo states)	0.540	0.430	0.460	0.613	1.579	-	5 617	8 343
Proposed (55 tempo states)	0.543	0.431	0.458	0.613	1.578	-	3 369	4 496
<i>GMM-BarTracker</i>								
Greek								
Original [12] (18 tempo states)	0.916	0.810	0.778	0.952	2.420	0.777	133 200	376 800
Proposed (35 tempo states)	0.956	0.850	0.935+	0.965	2.625	0.812	26 716	41 708
Indian								
Original [12] (18 tempo states)	0.799	0.684	0.613	0.845	1.988	0.476	133 200	376 800
Proposed (35 tempo states)	0.850+	0.737+	0.703	0.942+	2.415+	0.515	26 716	41 708
Turkish								
Original [12] (18 tempo states)	0.861	0.679	0.694	0.840	1.431	0.617	133 200	376 800
Proposed (35 tempo states)	0.877	0.689	0.732	0.877	1.575	0.632	26 716	41 708

Table 1: Performance of the original and proposed transition model on the *Ballroom*, *Hainsworth*, *SMC*, *Greek*, *Indian*, and *Turkish* dataset. The + symbol denotes significant ($p < 0.05$) improvement over the result in the row above, using a one-way analysis of variance (ANOVA) test of significance.

BarTracker, which has *a priori* a larger state space because it models (a) bars instead of beats and (b) eight rhythmic patterns. With the original *GMM-BarTracker*, processing a four-minute piece (12 000 frames at 50 fps), required remembering 1.60×10^9 state ids in the Viterbi algorithm, which needs 6.39 GB stored as 32-bit integers. In contrast, using the proposed model, only 0.32×10^9 states must be stored - a demand that can be met using 16-bit integers in only 0.64 GB of memory. With a MATLAB implementation on an Intel Core i5-2400 CPU with 3.1 GHz, we can therefore reduce the computation time for the *Turkish* dataset from 45.8 minutes to 4.2 minutes, including the computation of the audio features (which takes only 18 seconds). Additionally, as already shown in Experiment 1, we can further reduce the number of tempo states from 82 (the maximum number of tempo states as computed in Section 2.3.1) to 55 with the *RNN-BeatTracker*, with only marginal performance decrease. Compared to the original model, this implies a reduction of the numbers of states and transitions by factors of three and seven, respectively. Since in the proposed model most position states are needed to model lower tempi, the lower tempo limits mainly determine the size of the state space.

5. CONCLUSIONS

In this paper, we have proposed a new discretisation and tempo transition model that can be used as a drop-in replacement for variants of the *bar pointer model*. We have shown that our model outperformed the original one in 32 of 33 test cases, while substantially reducing space and time complexity. We believe that this is an important step towards lightweight, real-time capable, high-performance meter inference systems.

As part of future work, we plan to investigate whether changing tempo only at beat positions also stabilises the particle filter versions of the *bar pointer model* [15, 18], which would further facilitate reducing computational complexity.

6. ACKNOWLEDGMENTS

This work is supported by the European Union Seventh Framework Programme FP7 / 2007-2013 through the Austrian Science Fund (FWF) project Z159 and the GiantSteps project (grant agreement no. 610591). Thanks to Ingrid Abfalter for proofreading.

7. REFERENCES

- [1] S. Böck, F. Krebs, and G. Widmer. A multi-model approach to beat tracking considering heterogeneous music styles. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, 2014.
- [2] T. Collins, S. Böck, F. Krebs, and G. Widmer. Bridging the audio-symbolic gap: The discovery of repeated note content directly from polyphonic music audio. In *AES 53rd International Conference on Semantic Audio*, London, 2014. Audio Engineering Society.
- [3] M. Davies, N. Degara, and M. Plumbley. Evaluation methods for musical audio beat tracking algorithms. *Queen Mary University of London, Tech. Rep. C4DM-09-06*, 2009.
- [4] C. Drake and M. Botte. Tempo sensitivity in auditory sequences: Evidence for a multiple-look model. *Perception & Psychophysics*, 54(3):277–286, 1993.
- [5] S. Durand, J. Bello, D. Bertrand, and R. Gael. Downbeat tracking with multiple features and deep neural networks. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, 2015.
- [6] G. Dzhabazov. Towards a drum transcription system aware of bar position. In *Proceedings of the AES 53rd International Conference on Semantic Audio*, London, 2014.
- [7] M. Goto. AIST annotation for the RWC music database. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pages 359–360, Victoria, 2006.
- [8] F. Gouyon. *A computational approach to rhythm description-Audio features for the computation of rhythm periodicity functions and their use in tempo induction and music content processing*. PhD thesis, Universitat Pompeu Fabra, 2005.
- [9] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1832–1844, 2006.
- [10] S. Hainsworth and M. Macleod. Particle filtering applied to musical tempo tracking. *EURASIP Journal on Applied Signal Processing*, 2004:2385–2395, 2004.
- [11] A. Holzapfel, M. Davies, J. Zapata, J. Oliveira, and F. Gouyon. Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(9):2539–2548, 2012.
- [12] A. Holzapfel, F. Krebs, and A. Srinivasamurthy. Tracking the odd: Meter inference in a culturally diverse music corpus. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, 2014.
- [13] F. Korzeniewski, S. Böck, and G. Widmer. Probabilistic extraction of beat positions from a beat activation function. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, 2014.
- [14] F. Krebs, S. Böck, and G. Widmer. Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, Curitiba, 2013.
- [15] F. Krebs, A. Holzapfel, A. T. Cemgil, and G. Widmer. Inferring metrical structure in music using particle filters. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(5):817–827, 2015.
- [16] L. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [17] U. Şimşekli, O. Sönmez, B. Kurt, and A. Cemgil. Combined perception and control for timing in robotic music performances. *EURASIP Journal on Audio, Speech, and Music Processing*, 2012(1):1–20, 2012.
- [18] A. Srinivasamurthy, A. Holzapfel, A. Cemgil, and X. Serra. Particle filters for efficient meter tracking with Dynamic Bayesian networks. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, Málaga, 2015.
- [19] A. Srinivasamurthy and X. Serra. A supervised approach to hierarchical metrical cycle tracking from audio music recordings. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5217–5221, Florence, 2014.
- [20] N. Whiteley, A. Cemgil, and S. Godsill. Bayesian modelling of temporal structure in musical audio. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, Victoria, 2006.
- [21] N. Whiteley, A. Cemgil, and S. Godsill. Sequential inference of rhythmic structure in musical audio. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages IV–1321, Honolulu, 2007.
- [22] J. Zapata, M. Davies, and E. Gómez. Multi-feature beat tracking. *IEEE/ACM Transactions on Audio, Speech and Language Processing*, 22(4):816–825, 2014.

AUTOMATIC HANDWRITTEN MENSURAL NOTATION INTERPRETER: FROM MANUSCRIPT TO MIDI PERFORMANCE

Yu-Hui Huang^{◊*}, Xuanli Chen^{◊*}, Serafina Beck[†], David Burn[‡], and Luc Van Gool^{◊‡}

[◊]ESAT-PSI, iMinds, KU Leuven [†]Department of Musicology, KU Leuven [‡]D-ITET, ETH Zürich

{yu-hui.huang, xuanli.chen, luc.vangool}@esat.kuleuven.be, {serafina.beck, david.burn}@art.kuleuven.be

*equal contribution

ABSTRACT

This paper presents a novel automatic recognition framework for hand-written mensural music. It takes a scanned manuscript as input and yields as output modern music scores. Compared to the previous mensural Optical Music Recognition (OMR) systems, ours shows not only promising performance in music recognition, but also works as a complete pipeline which integrates both recognition and transcription.

There are three main parts in this pipeline: i) region-of-interest detection, ii) music symbol detection and classification, and iii) transcription to modern music. In addition to the output in modern notation, our system can generate a MIDI file as well. It provides an easy platform for the musicologists to analyze old manuscripts. Moreover, it renders these valuable cultural heritage resources available to non-specialists as well, as they can now access such ancient music in a better understandable form.

1. INTRODUCTION

Cultural heritage has become an important issue nowadays. In the recent decades, old manuscripts and books have been digitalized around the world. As more and more libraries are carrying out digitalization projects, the number of manuscripts increases exponentially every day. The texts in these manuscripts can be further processed using Optical Character Recognition (OCR) techniques while the music notes can be processed by Optical Music Recognition (OMR) techniques. However, due to the nature of the manuscript, the challenges of OMR and OCR have to be addressed differently. For example, OMR has to deal with different types of notations from different time periods, such as Chant notation used throughout the medieval and the Renaissance periods while white mensural notation used during the Renaissance. Even within the same period, music symbols vary in different geographical areas [13]. In



© Yu-Hui Huang^{◊*}, Xuanli Chen^{◊*}, Serafina Beck[†], David Burn[‡], and Luc Van Gool^{◊‡}.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Yu-Hui Huang^{◊*}, Xuanli Chen^{◊*}, Serafina Beck[†], David Burn[‡], and Luc Van Gool^{◊‡}. "Automatic Handwritten Mensural Notation Interpreter: from Manuscript to MIDI Performance", 16th International Society for Music Information Retrieval Conference, 2015.



Figure 1: The overview of our framework. (a) Original image after ROI selection. (b) After preprocessing. (c) Symbol segmentation. (d) Transcription results.

addition to the semantic characteristic, OMR has the additional problem as OCR of having to cope with the physical condition of historical documents [15].

While several OMR systems exist for ancient music scores in white mensural notation, most of them target at printed scores. To name a few, Aruspix [3] is an open source OMR software targeting those ancient printed scores; Pugin et al. utilized the Hidden Markov Models to recognize the music symbols and to incorporate the pitch information simultaneously. A comparative study made by Pugin et al. [13] shows that Aruspix has better performance on selected printed books than Gamut [11], which is another OMR software based on the Gamera [9] open-source document analysis framework. Gamut first segments the symbols based on the result after staff lines removal, and classifies it using kNN classifier.

Calvo-Zaragoza et al. [5] proposed an OMR system without removing the staff lines. They utilized histogram analysis to segment the staves as well as different music symbols, and classified by cross-correlating templates. Their method achieves averagely an extraction rate of 96%

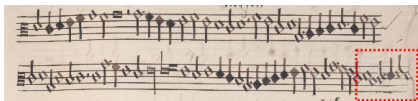


Figure 2: A regular case happens at the end of each voice: due to a lack of space, the writer extends the staff lines a little bit (red dashed box) and squeezes the remaining symbols on.

on the Archivo de la Catedral de Malaga collection which has a certain printing style.

In addition to the physical condition of the manuscripts, the substantial difference in style between writers renders OMR challenge. One and the same symbol can appear quite differently, depending on the writer. Moreover, the symbols sometimes are written too close to each other which increases the difficulty of symbol segmentation. This usually happens at the end of each voice as the writer wants to finish on the same line instead of adding a new one. In such cases, they usually elongate the staff lines manually in order to add more symbols, see e.g. Figure 2. Such cases increase the difficulty to apply OMR on these handwritten manuscripts in a systematic and consistent manner.

Similar to Gamut, we remove staff lines to detect the symbols, but differently, we employ the Fisher Vector [12] representation to describe images and Support Vector Machines (SVM) to classify them. With relatively less training data compared to others, our OMR system is able to recognize the symbols from different writers with high accuracy.

In contrast to the modern music (the music from the so-called Common practice period), the music notation up to the Renaissance is much different in appearance. Therefore, transcription from an expert is required to further process the data. Our goal therefore was to design and implement a system that automatically transcribes such music for users who lack the expert knowledge about these early manuscripts. In particular, our system is able to automatically transcribe most of contents in mensural music pieces as shown in Figure 1. We propose a new OMR system which not only recognizes the handwritten music scores but also transcribes it from white mensural notation to the modern notation. The modern notation is then encoded into MIDI files. The overall pipeline is described in Figure 3. In addition to provide a user friendly platform for the musicologists to analyze the music from old manuscripts, our system renders these valuable cultural heritage resources to non-specialists as well. Compared to most OMR system, the playable MIDI files in our system help people without any music knowledge access those ancient music.

The remaining of this paper is structured as the following. Section 2 describes the image preprocessing steps. In Section 3, we introduce the core part of the OMR system, music symbol recognition. The transcription to modern notation is explained in Section 4. Experimental settings

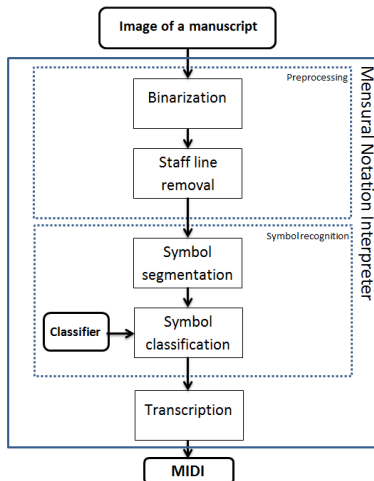


Figure 3: The overall scheme of our framework.

and results are shown in Section 5. Section 6 concludes the paper.

2. PREPROCESSING

Following typical OMR pipelines, we start from a preprocessing step. It consists of two parts, namely binarization and stave detection.

2.1 Binarization

In some collections of manuscripts, each scanned image comes up with a color check and a ruler aside the main manuscript. In order to achieve a good quality, the non-music parts need to be removed during the binarization. Given a high resolution scanned image of a music manuscript, the boundaries of the page are first detected by histogram analysis of pixel intensity in gray-scaled image. Thresholds are set to the horizontal and vertical histograms to segment the x- or y-axis into two parts: the page part containing the staves and the background parts together with the color check and the ruler. Because the Region of Interest (ROI) refers to the page part here, which contains much higher intensity of grey values compared to the black background. Based on this fact, with properly chosen threshold, ROI could be well selected. The result is shown in Figure 1a.

For those manuscripts containing colored initials or decorations, we apply K-means clustering under the Lab color space in order to filter out some colored non-music elements after cropping out the color check and the ruler. In the experiments we put K to the value 2, and successfully cluster the manuscript into two groups, the elements

with red color and the others containing the staff. We select the red group to build a mask to remove those non-music regions from the manuscript. After that, we then apply Otsu threshold to do the binarization. For simplicity, we will focus on a specific style, generating to other styles of manuscripts will be considered in the future work. Figure 1b shows an example result after these preprocessing steps applied.

2.2 Stave detection and staff lines removal

The stave in mensural notation are mostly composed of five lines. Based on this assumption, we use the stave detection program from [16]. Timofte et al. utilized dynamic programming to retrieve the patterns of five lines in order to detect the stave. While detecting the staff lines, the parameters of staff line thickness and space between two staff lines are optimized at the same time. Figure 1b shows the result after staff removal.

3. SEGMENTATION AND CLASSIFICATION

With the preprocessing steps of the previous section having been completed, we obtain binarized images without staff lines. In this section, we first describe how the symbols are segmented and then how the classification of the individual, segmented symbols works.

3.1 Segmentation

Given a binarized image without staff lines (Figure 4a), we employ the connected component analysis to separate different symbols. However, the symbols touching the staff line in the original manuscript may become separate after staff removal. As the Figure 4b shows, a semibreve or a minim may be separated into two parts. To solve this problem, we set up several heuristic rules to combine the parts of such broken symbols. For example, we observe that some overlapping or close neighbouring boxes detected with similar width could be merged into one individual symbol. Therefore we merge neighbouring boxes in this case. Yet, this procedure might be risky in that two close parts coming from different symbols may get erroneously merged as well. To tackle that, we set up a width threshold for merging boxes, i.e. if the box width is more than two times of the space between two staff lines, the two boxes will not be merged. The final result is shown in Figure 4c.

Moreover, in order to distinguish the lyrics from the music symbols, we use the stave region detected from the previous section as a mask to filter out those non-music symbols.

3.2 Classification

In order to train the classifier, we manually annotate the image of each music symbol by drawing the bounding box around it using the image annotation tool [4] from the original manuscript. Because the bounding boxes may differ from each other in size, for each cropped symbol I , we

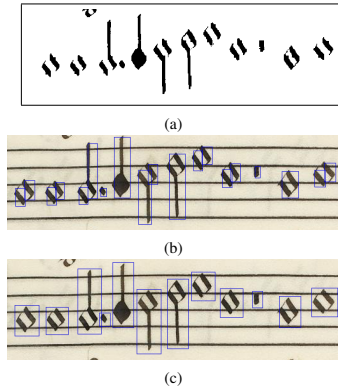


Figure 4: (a) After staff removal, some symbols become separated because some strokes touching the staff line are removed as well. (b) The result of applying connected component analysis on Figure 4a. (c) The result after applying heuristic rules to combine the broken symbols from Figure 4b.

first normalize the image to the same height, which is determined by ensuring enough SIFT [10] features could be extracted to form the Fisher Vector representation [12].

For training, we use a Gaussian Mixture Model (GMM) with $K = 128$ components as the generative model for the patch descriptors. To estimate the parameters of the GMM, we obtain 10000 sample descriptors by applying dense SIFT in all the images from the training data, and reduce them from $D = 128$ -d to $D = 64$ -d using PCA. Then, the mean, variance and weight of each Gaussian component are estimated using the Expectation Maximization algorithm. The final fisher vector for an image I has dimension of $2KD = 16384$. This vector is then signed-squared-rooted and l_2 normalized. We follow the procedures in [7] to obtain the improved fisher vector, that is after pooling all the vectors of the training data, we apply the square rooting again to normalize. With bunch of vectors from training data, we train the classifier using linear SVM. We train the multi-class classifier with one versus one strategy and select the class with the highest probability.

During testing, we already obtained the bounding box information of each music symbol following the previous steps. To avoid the effect caused by binarization and staff removal, we extract the symbol again directly from the original colored image using the same coordinates given by the bounding box. Hereby, we would like to remind that the preprocessing steps are for symbol segmentation, while both training and testing patches are extracted from the original manuscript. Each segmented symbol is described in Fisher Vector representation in a similar way as we did for the training data. Then we use the trained multi-

class classifier to predict its class.

In our implementation, we used the VLFeat library [17] for the Fisher Vector and SIFT implementations, and libsvm [6] with linear kernel and default settings for the Support Vector Machine.

3.3 Pitch detection and channel separation

The pitch information is essential for transcription, and the pitch level is determined by the relative position of the note and the clef to the staff. After the music symbol is extracted and classified, we follow the post-processing steps described in [14] to retrieve the pitch level.

We divide the group of notes into two groups according to their stems. For the group of notes with stems, we perform histogram analysis to extract the y position of the stem, so that this can be used to localize the center of the note head. The detail of the histogram analysis is as following: we first project all the horizontal pixels onto the y-axis and then set up a threshold to separate the stem and the note head, by employing the fact that note head part has higher intensity of pixels than the stem. For the group of none-stem notes, we simply compute the middle point, departing from the highest and the lowest points of the symbol.

For clefs, the point of relevance is much easier to locate, since they can only be situated on staff lines. We simply determine the middle point of two squares from clef c and of the two dots or blobs from the right part of clef f, while we locate the center of the blob for clef g. For key signatures, we only encounter the case of flat, as the sharp is rare in the dataset we use. We adopt a similar strategy to that of the notes to locate the center of the blob for the flat symbol. With the relative position of the extracted symbol calculated, we connect this information to the staff line position in order to determine the pitch level of the corresponding symbol.

In the case of choirbooks, there are always several voices within one page of a manuscript in our dataset. Thus, in order to transcribe the music correctly, we need to recognize these different voices. As each voice ends with barlines, we use this as a criterion to separate different voices. After a barline is detected, we switch the notes detected afterwards to another channel.

4. TRANSCRIPTION

We aim at transcribing mensural music scores into modern notations. This will render the music accessible to a far larger group of people, also because much of this music has not even been published. The tool is also valuable for musicologists, because it takes over the time consuming manual transcription work. Instead, they can spend their time on the actual music analysis. With the vast digital manuscript collections of libraries that are being made available daily, the transcription tool makes it a lot easier to establish concordances. Also, printed and often not published transcriptions are sometimes hard to get by, so this tool means generally a big improvement of accessi-

bility of transcriptions. Moreover, with all the available software libraries nowadays, such as *music21* [8] which is also used in our work, MIDI files could be generated directly from modern notation scores. Therefore the mensural script could be more easily accessed by general public.

4.1 Transcription rules

There are several difficulties in transcribing mensural music. Apart from notational challenges like ligatures and coloration, which are not supported yet, the main challenge of mensural music transcription is how to translate the mensuration, or time signature. In contrast to modern music, the time signature defines not only how long one measure is, but also defines how to divide a certain note.

There are four kinds of notes that can be divided in different ways. The division of maxima into longa is called *modus maximarum* or *modus maior*. From longa into breves, it is called *modus*. From breves into semibreves, it is called *tempus*. And from semibreves into minims, it is called *prolatio*. For all of these four divisions, depending on whether they are *perfect* or not, either a *ternary* or *binary* division is possible. If a note is divided in a *perfect*, i.e. *ternary* way, it will be divided into three sub-class notes. If one note however is divided in an *imperfect*, i.e. *binary* way, it will be divided into two sub-class notes. For example, in a case of *perfectum*, a longa will be divided into three breves, while in a case of *imperfectum*, the *modus* specifies that one longa has to consist of two breves. This rule also applies to the other three transcription pairs.

The temporal length of one breve in mensural music defines the length of one measure in modern music. In the normal case (i.e. without scaling of temporal length), the length of a semiminim equals that of a modern quarter note. Because a semiminim cannot be affected by the rules of *perfect/imperfect* for its division into its sub-class fusa (i.e. a quaver in modern notation), we are able to calculate the actual length of a breve, by treating semiminims as a unit. For instance, if *tempus* and *prolatio* (which refer to the semibreve-minim division and breve-semibreve, respectively), are both *perfect*, then one breve will be divided into three semibreves, and each semibreve will in turn be divided into three minims. As a result, one breve is divided into nine semiminims. If we treat one semiminim as a beat, then the corresponding time signature would be 9/4.

In addition, there is a variant version of mensuration symbols, these are the time signatures with a vertical line through the original symbol, usually called *cut-signs*. They imply a reduction of all the temporal values, of notes and rests, by a factor of two. In other words, with *cut-sign*, the playing speed of the music will be twice as faster. Note that most mensural music is rather slow compared to contemporary music. Beside the *cut-signs*, we also provide a parameter to artificially scale the speed of playing. In order to achieve that, we only need to change the mapping relationship between the mensural notes and the modern ones. For instance, in no-scaling cases, a semiminim is mapped to a quarter. If we speed up the music by two times, we just need to map semiminim to a quaver, which is half the

length of a quarter. In this case, one should adapt the time signature accordingly.

4.2 Implementation details

Given the aforementioned observations, the analysed mensural music can be encoded into modern music. In our pipeline, we first check the mensuration of the music piece. Taking into consideration that the mensuration might change at any time during the piece, this step should be repeated any time during the process. If there have been any changes, we apply the reduction ratio to the music afterwards. After this, we can determine the mapping relation between the semiminim and modern music notes and calculate the modern time signature according to the duration of one breve in the transcription. With determined time signatures and basic mapping relationships established, we can transcribe each element into modern musical notation, note by note and rest by rest. If the division is only binary or imperfect, we can directly transcribe the mensural music to modern music. We are still working on the transcription techniques for the perfect divisions, which include a lot more exceptions that can still present challenges. Once musical symbol recognition are ready, all we need to do is to carefully encode these symbols. One should be especially aware of the possibility that clefs and/or time signatures change in the middle of a piece. For this step, we chose the framework offered by *music21* [8] to encode the music information, because it offers an automatic parsing library and APIs towards visualization and MIDI output. The different voices in the original music sheet are encoded into different 'part objects' in this framework, while the whole piece is treated as a 'stream' object. Another thing that needs to be taken care of is the *punctus divisionis* sometimes appearing in a *perfect* division, which looks exactly like a normal dot with the function of prolonging note values, but instead of prolonging, the *punctus divisionis* functions as a kind of barline. Whether or not we are dealing with this kind of dot, should be established from the note durations directly preceding and following it.

5. EXPERIMENTAL RESULTS

5.1 Dataset and evaluation

We evaluate our pipeline on the Alamire collection which includes manuscripts of various writers in several books. Depending on the sources, those manuscripts are in high resolution from 7200x5400 to 10500x7400 pixels. For training, we randomly select the manuscripts from the following books: *Vienna, Österreichische Nationalbibliothek (VienNB)*, *MS Mus.* 15495, 15497, 15941, 18746; *Brussels, Koninklijke Bibliotheek (BrusBR)* *Ms.* 228, and *IV.922* [1]. We use the image annotation tool made by Kläser [4] to manually draw the bounding box around each symbol and to annotate the corresponding information. In total we have about 2800 samples for training over 33 classes. The classes include the notes, rests, key signature

Book	MunBS F	LonBLR	MS 72A
N	839	1313	1636
R_{ext}	85.73%	94.36%	90.25%

Table 1: Symbol extraction result on three books.

(flat), most of the frequent time signatures and other symbols such as barlines and custos. The testing data comes from different books, without any overlap with the training data: *Munich, Bayerische Staatsbibliothek, Mus. MS. F (MunBS F)* [2]; *London, British Library MS Royal 8G.vii (LonBLR)*, and *'s-Hertogenbosch, Archief van de Illustre Lieve Vrouwe Broederschap, MS. 72A (MS 72A)* [1]. In total, there are about 3700 samples for testing. In our evaluation, we report the result of classification and segmentation separately.

5.2 Symbol segmentation

We follow the evaluation process in [5]. The extraction rate is defined as $R_{ext} = \frac{M_e}{T}$, where M_e is the number of music symbols extracted and T is the total number of music symbols within the manuscript. Table 1 shows the symbol segmentation results on three collections where N is the total number of symbols per book. Most false negatives of detection come from custodies, as they are often over segmented into several parts after staff removal. Some of the other false negatives come from the symbols on the sixth staff line, below or above the stave, causing the symbols above or below the stave not correctly extracted. Moreover, the ornate capitals in front of the piece may distract the detection especially on the MunBS F collection. Unlike the colored initials in LonBLR, the black initial makes the separation of symbols more difficult. These issues are being solved and will be addressed in the future work.

5.3 Symbol classification

To evaluate the classification step, we first correct the segmentation errors from the last step as Figure 1c shows, and then use prediction accuracy to evaluate the classification. Table 2 presents the classification result on the same collections. The accuracy reaches 98 % on the LonBLR and the MS 72A collections, and 95 % on the MunBS F collection. After analysis, we found the typical error for the MS 72A collection is the misclassification of a breve rest as a colored breve. In MunBS F, most of the classification errors are from the semibreve notes which are mistakenly classified as points. Some incidents are caused by similar symbols, such as the note fusa recognized as semiminim and the note maxima classified as longa. The reason might be found in the imbalanced training samples in our training set. As some symbols do not happen appear so often such as the note maxima and time signatures, they are less present in the set. It makes the training collection more challenging if one wants to avoid this issue.

With limited training data, the use of the Fisher Vectors and SVMs yields a promising classification perfor-

Book	MunBS F	LonBLR	MS 72A
Accuracy	95.52%	98.83%	98.94%

Table 2: Classification result on three books.

mance on handwritten symbols from different writers. As the manually annotated training data is hard to obtain, our method shows an obvious advantage compared to earlier alternatives.

6. CONCLUSION

In this paper, we presented a framework to automatically analyse and transcribe handwritten mensural music manuscripts. The inclusion of the transcription part not only provides the musicologists with a simple platform to more efficiently study those manuscripts, but also assists music amateurs to explore and enjoy this ancient music. Moreover, the MIDI-output feature offers the public at large easy and convenient access to these musical treasures.

We have collected a dataset of handwritten mensural notation symbols from different books for evaluation. We believe it is fair to claim that our symbol segmentation attains good performance. The classification based on the Fisher Vector representation and SVMs achieves very high classification rate on handwritten symbols. Furthermore, we implemented an accurate transcription mechanism which embeds musicological information.

We plan to extend this work by enabling counterpoint checking so that mistakes in original music manuscripts can be pointed out to the musicologists easily. In addition, we intend to implement scribe identification in our system (an early module for that is ready) to assist authorship identification.

7. ACKNOWLEDGMENTS

We are grateful to Alamire Foundation for their support and we would like to thank Lieselotte Bijens, Lonne Maris, Karen Schets and Tim Van Thuyne for their help on symbol annotations. The work is funded by the Flemish IWT/SBO project: New Perspectives on Polyphony, Alamire's musical legacy through high-technology research tools.

8. REFERENCES

- [1] IDEM. <http://elise.arts.kuleuven.be/alamire/>. Accessed: 2015-04-23.
- [2] Munich, Bayerische Staatsbibliothek, Handschriften-Inkunabelsammlung, Musica MS F. <http://www.digitale-sammlungen.de/>. Accessed: 2015-04-23.
- [3] Aruspix project. <http://www.aruspix.net/>, 2008. Accessed: 2015-04-23.
- [4] Image annotation tool with bounding boxes. <http://lear.inrialpes.fr/people/klaeser/software>, 2010. Accessed: 2015-04-23.
- [5] J. Calvo-Zaragoza, I. Barbancho, L. J. Tardn, and A. M. Barbancho. Avoiding staff removal stage in optical music recognition: application to scores written in white mensural notation. *Pattern Analysis and Applications*, pages 1433–7541, 2014.
- [6] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2:27:1–27:27, 2011. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [7] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *British Machine Vision Conference*, 2011.
- [8] M. S. Cuthbert and C. Ariza. music21: A toolkit for computer-aided musicology and symbolic music data. In *Proceedings of the ISMIR 2010 Conference*, 2010.
- [9] M. Droettboom, G. S. Chouhury, and T. Anderson. Using the gamera framework for the recognition of cultural heritage materials. In *Joint Conference on Digital Libraries : Association for Computing Machinery*, 2002.
- [10] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 2004.
- [11] K. MacMillan, M. Droettboom, and I. Fujinaga. Gamera: Optical music recognition in a new shell. In *Proceedings of the International Computer Music Conference*, 2002.
- [12] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *Proceedings of the 11th European Conference on Computer Vision*, 2010.
- [13] L. Pugin and T. Crawford. Evaluating omr on the early music online collection. In *Proceedings of the ISMIR 2013 Conference*, 2013.
- [14] L. Pugin, J. Hockman, J. A. Burgoyne, and I. Fujinaga. Gamera versus aruspix – two optical music recognition approaches. In *Proceedings of the ISMIR 2008 Conference*, 2008.
- [15] C. Ramirez and J. Ohya. Symbol classification approach for omr of square notation manuscripts. In *Proceedings of the ISMIR 2010 Conference*, 2010.
- [16] R. Timofte and L. Van Gool. Automatic stave discovery for musical facsimiles. In *Asian Conference on Computer Vision*, 2012.

- [17] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008.

INFINITE SUPERIMPOSED DISCRETE ALL-POLE MODELING FOR MULTIPITCH ANALYSIS OF WAVELET SPECTROGRAMS

Kazuyoshi Yoshii¹ Katsutoshi Itoyama¹ Masataka Goto²

¹Graduate School of Informatics, Kyoto University, Japan

²National Institute of Advanced Industrial Science and Technology (AIST), Japan

{yoshii, itoyama}@kuis.kyoto-u.ac.jp m.goto@aist.go.jp

ABSTRACT

This paper presents a statistical multipitch analyzer based on a source-filter model that decomposes a target music audio signal in terms of three major kinds of sound quantities: pitch (fundamental frequency: F0), timbre (spectral envelope), and intensity (amplitude). If the spectral envelope of an isolated sound is represented by an all-pole filter, linear predictive coding (LPC) can be used for filter estimation in the linear-frequency domain. The main problem of LPC is that although only the amplitudes of harmonic partials are reliable samples drawn from the spectral envelope, the whole spectrum is used for filter estimation. To solve this problem, we propose an *infinite superimposed discrete all-pole (iSDAP)* model that, given a music signal, can estimate an appropriate number of superimposed harmonic structures whose harmonic partials are drawn from a limited number of spectral envelopes. Our nonparametric Bayesian source-filter model is formulated in the log-frequency domain that better suits the frequency characteristics of human audition. Experimental results showed that the proposed model outperformed the counterpart model formulated in the linear frequency domain.

1. INTRODUCTION

Statistical modeling of music audio signals based on machine learning techniques is a hot topic in the field of music signal analysis. In particular, nonnegative matrix factorization (NMF) has often been used for multiple fundamental frequency (F0) estimation (multipitch analysis) and source separation [1–4, 7, 14, 15, 17, 21–26]. The standard NMF approximates a nonnegative spectrogram (matrix) as the product of two nonnegative matrices: a set of basis spectra and a set of the corresponding activations. An efficient multiplicative-updating (MU) algorithm was proposed for minimizing a cost function that measures the approximation error [18]. This was later found to be maximum likelihood estimation of a particular probabilistic model [5].

Statistical source-filter models, which were inspired by the simplified model of the speech production mechanism,

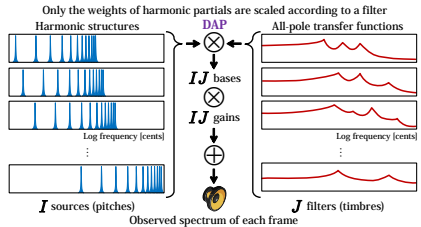


Figure 1. Overview of infinite superimposed discrete all-pole (iSDAP) modeling: We take the infinite limit as both the numbers of sources and filters, I and J , go to infinity.

have often been proposed for representing musical instrument sounds [7, 14, 25]. The pitches and timbres of musical instrument sounds are well characterized by fine structures (sources) and spectral envelopes (filters) in the frequency domain. Since the human auditory system is sensitive to spectral peaks and formants, the spectral envelope of each frame is usually modeled by an all-pole frequency transfer function (frequency response of an autoregressive (AR) filter) [14]. A classical method of all-pole spectral envelope estimation called linear predictive coding (LPC) [16] corresponds to maximum likelihood estimation of a particular probabilistic model under a strong assumption that source signals have the flat spectrum (white noise).

The composite autoregressive (CAR) modeling [17] is a promising statistical approach that overcomes the limitation of classical source-filter modeling in the framework of NMF. A given audio spectrogram is decomposed into specified numbers of fine structures (sources) and spectral envelopes (filters). A key feature of this approach is that source spectra themselves can be estimated (not limited to white noise) at the same time as all-pole spectral envelope estimation. The probabilistic interpretation of source-filter NMF makes it possible to formulate a nonparametric Bayesian extension called *infinite* CAR (iCAR) modeling that can automatically choose the appropriate numbers of sources and filters according to a given audio spectrogram [26]. Another useful extension is to restrict source spectra to harmonic structures by using parametric functions [26]. The F0s of source spectra can be estimated in a principled maximum-likelihood framework.

Conventional methods of source-filter NMF including CAR [7, 14, 17, 25, 26] have two major problems as follows:

© Kazuyoshi Yoshii, Katsutoshi Itoyama, Masataka Goto. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Kazuyoshi Yoshii, Katsutoshi Itoyama, Masataka Goto. "Infinite Superimposed Discrete All-pole Modeling for Multipitch Analysis of Wavelet Spectrograms", 16th International Society for Music Information Retrieval Conference, 2015.

1. All the frequency bins are taken into account for spectral envelope estimation although only the amplitudes of harmonic partials can be regarded as reliable samples from spectral envelopes.
2. Linear-frequency spectrograms given by short-time Fourier transform (STFT) are used for all-pole modeling although log-frequency spectrograms given by wavelet or constant-Q transform better suit the frequency characteristics of human audition.

To solve these problems, we propose a new statistical approach to source-filter NMF called *infinite superimposed discrete-all-pole* (iSDAP) modeling. Our approach is based on a well-known technique called *discrete all-pole* (DAP) modeling [8] that takes into account only the peaks of harmonic partials for spectral envelope estimation. To deal with polyphonic music audio signals, however, we need to separate individual harmonic structures and estimate their FOs (positions of discrete harmonic partials). A major contribution of this study is to extend the DAP modeling for dealing with an arbitrary number of superimposed harmonic structures in a similar way to the iCAR modeling. This enables us to decompose a log-frequency spectrogram into appropriate numbers of pitches (FOs), timbres (spectral envelopes), and their volumes by leveraging the frequency-scale-free characteristics of the DAP modeling.

2. RELATED WORK

This section reviews probabilistic models of source-filter decomposition, NMF, and source-filter NMF as a basis of formulating the iSDAP model. Most conventional models are formulated in the linear frequency (STFT) domain.

2.1 Linear Predictive Coding (All-pole Modeling)

The linear predictive coding (LPC) [16] is a signal modeling method that can be used for estimating the spectral envelope of an observed spectrum. The underlying assumption is that the corresponding audio signal $\mathbf{x} = \{x_m\}_{m=-\infty}^{\infty}$ (a local signal $\{x_m\}_{m=1}^M$ is infinitely repeated) follows a P -order autoregressive (AR) process as follows:

$$x_m = -\sum_{p=1}^P a_p x_{m-p} + s_m \quad i.e., \quad \sum_{p=0}^P a_p x_{m-p} = s_m, \quad (1)$$

where $\mathbf{a} = [a_0, \dots, a_P]^T$ is a vector of AR coefficients ($a_0 = 1$) and $\{s_m\}_{m=1}^M$ is a set of prediction errors. Eq. (1) can be interpreted in terms of source-filter modeling, *i.e.*, when \mathbf{x} is a speech signal, \mathbf{s} is an excitation signal generated by the vocal cords (source) and \mathbf{a} represents the resonance characteristics of the vocal tract (filter).

Eq. (1) can be regarded as a linear system (governed by \mathbf{a}) that takes \mathbf{s} as input and then gives \mathbf{x} as output. Since Eq. (1) is a convolution of \mathbf{a} with \mathbf{x} , we can say

$$A(z)X(z) = S(z) \quad i.e., \quad X(z) = S(z)F(z), \quad (2)$$

where $X(z)$ and $S(z)$ are the z -transforms of \mathbf{x} and \mathbf{s} , respectively, which are given by

$$X(z) = \sum_{m=-\infty}^{\infty} x_m z^{-m} \quad \text{and} \quad S(z) = \sum_{m=-\infty}^{\infty} s_m z^{-m}, \quad (3)$$

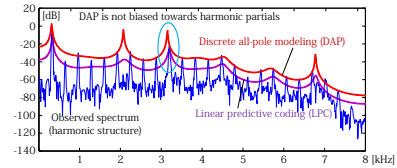


Figure 2. Spectral envelopes estimated by LPC and DAP.

and $F(z) \stackrel{\text{def}}{=} \frac{1}{A(z)}$ is an all-pole transfer function given by

$$F(z) = \frac{1}{A(z)} = \frac{1}{\sum_{p=0}^P a_p z^{-p}}. \quad (4)$$

Letting $2\pi \frac{m}{M} = \omega_m$ and substituting $z = e^{i\omega_m}$ into Eq. (2), we get the Fourier-transform representation as follows:

$$X(e^{i\omega_m}) = S(e^{i\omega_m})F(e^{i\omega_m}), \quad (5)$$

where $\{X(e^{i\omega_m})\}_{m=1}^M$ is the complex spectrum of the observed signal \mathbf{x} , $\{S(e^{i\omega_m})\}_{m=1}^M$ is that of the source signal \mathbf{s} , and $\{F(e^{i\omega_m})\}_{m=1}^M$ is the frequency characteristics of the all-pole transfer function.

The goal of LPC is to estimate a set of AR coefficients \mathbf{a} under a strong unrealistic assumption that the source signal \mathbf{s} is Gaussian white noise. This means that $S(e^{i\omega_m})$ is complex Gaussian-distributed as follows:

$$S(e^{i\omega_m}) \sim \mathcal{N}_c(0, \sigma^2), \quad (6)$$

where σ^2 is the power of the white spectrum of the source signal \mathbf{s} . Using Eq. (5) and Eq. (6), we get

$$X(e^{i\omega_m}) \sim \mathcal{N}_c(0, \sigma^2 |F(e^{i\omega_m})|^2). \quad (7)$$

Letting $X_m = |X(e^{i\omega_m})|^2$ and $F_m = |F(e^{i\omega_m})|^2$, we briefly rewrite Eq. (7) as follows:

$$X_m \sim \text{Exponential}(\sigma^2 F_m), \quad (8)$$

where $\{X_m\}_{m=1}^M$ is the *power* spectrum of the observed signal \mathbf{x} and $\{F_m\}_{m=1}^M$ is the spectral envelope of $\{X_m\}_{m=1}^M$, as shown in Figure 2. Eq. (8) defines the probabilistic model of LPC. $\{F_m\}_{m=1}^M$ (*i.e.*, \mathbf{a}) and σ^2 can be estimated in a maximum-likelihood manner [16].

The main problem of LPC is that if we analyze a pitched sound derived from a periodic source signal (*e.g.*, vibration of strings), the estimated envelope $\{F_m\}_{m=1}^M$ loosely fits the observed spectrum $\{X_m\}_{m=1}^M$ and its peaks (formants) tend to be biased to the positions of harmonic partials. This is because all M frequency bins are used for all-pole modeling although in reality only the amplitudes of harmonic partials can be considered to be reliable samples from the spectral envelope.

2.2 Discrete All-pole Modeling

The discrete all-pole (DAP) modeling [8] is a well-known spectral envelope estimation method that was proposed for solving the problem of LPC. Since DAP is an extension of LPC, the probabilistic model of DAP has the same form as Eq. (8). A key feature of DAP is that Eq. (8) is defined over only a partial set of frequency bins, Ω , as follows:

$$X_m \sim \text{Exponential}(\sigma^2 F_m) \quad m \in \Omega, \quad (9)$$

where if $\Omega = \{1, \dots, M\}$, DAP reduces to LPC. To estimate the spectral envelope of a harmonic spectrum, we can take into account only the discrete frequencies of harmonic partials. The estimated envelope passes close to the peaks of harmonic partials (Figure 2). To maximize the likelihood given by Eq. (9), an efficient algorithm was proposed for alternately optimizing α and σ^2 [8]. It was later found as a multiplicative updating algorithm [1, 14].

The main limitation of DAP is that the F0 and its overtones of an observed spectrum $\{X_m\}_{m=1}^M$ should be given in advance for defining a set of discrete frequencies to be considered, Ω . To analyze polyphonic music audio signals consisting of superimposed harmonic structures, we need to separate harmonic structures and estimate their F0s.

2.3 Composite Autoregressive Modeling

The composite autoregressive (CAR) modeling [17] is a variant of source-filter NMF that is used for decomposing a linear-frequency mixture spectrogram into I fine structures (sources) and J spectral envelopes (filters), as shown in Figure 3. Let \mathbf{X} be an $M \times N$ power spectrogram, where M is the number of frequency bins and N is the number of frames. The nonnegative matrix \mathbf{X} is factorized into three kinds of “factors” \mathbf{S} , \mathbf{F} , and \mathbf{H} as follows:

$$X_{mn} \approx \sum_{i=1}^I \sum_{j=1}^J S_{im} F_{jm} H_{nij} \stackrel{\text{def}}{=} Y_{mn}, \quad (10)$$

where $\{S_{im}\}_{m=1}^M$ is the linear-frequency power spectrum of source i , $\{F_{jm}\}_{m=1}^M$ is that of filter j , and H_{nij} is the gain of a pair of source i and filter j at frame n . All these variables should be estimated from \mathbf{X} .

2.3.1 Original Formulation

The probabilistic model of CAR can be formulated by precisely modeling source signals in a statistical manner. To avoid the unrealistic assumption of LPC that each source signal is Gaussian white noise (Eq. (6)), we assume

$$S_i(e^{i\omega_m}) \sim \mathcal{N}_c(0, S_{im}), \quad (11)$$

where $\{S_i(e^{i\omega_m})\}_{m=1}^M$ is the complex spectrum of source i . Using Eq. (5) and Eq. (11), we get

$$X_{ijmn}(e^{i\omega_m}) \sim \mathcal{N}_c(0, S_{im} F_{jm} H_{nij}), \quad (12)$$

where $\{X_{ijmn}(e^{i\omega_m})\}_{m=1}^M$ is a *latent* complex spectrum generated from source i and filter j at frame n . Using the reproducing property of the Gaussian and Eq. (10), we get

$$X_{mn}(e^{i\omega_m}) \sim \mathcal{N}_c(0, Y_{mn}), \quad (13)$$

where $\{X_{mn}(e^{i\omega_m})\}_{m=1}^M$ is the *observed* complex spectrum at frame n . Eq. (13) is equivalent to

$$X_{mn} \sim \text{Exponential}(Y_{mn}), \quad (14)$$

where $\mathbb{E}[X_{mn}] = Y_{mn}$ is satisfied and $\{X_{mn}\}_{m=1}^M$ and $\{Y_{mn}\}_{m=1}^M$ are the *power* spectra of frame n .

This means that the Itakura-Saito (IS) divergence is theoretically justified as a cost function that evaluates the error between X_m and Y_m in Eq. (10) [17]. In general, however, optimization algorithms tend to get stuck in bad local minima because the IS divergence is not convex w.r.t. Y_{mn} .

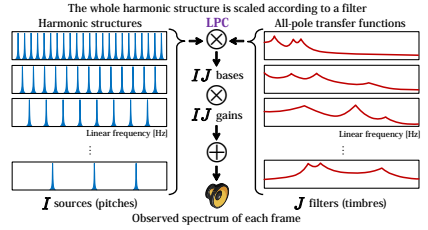


Figure 3. Overview of composite autoregressive (CAR) modeling defined in the linear frequency domain.

2.3.2 Several Extensions

Another probabilistic model of CAR was proposed by using the Kullback-Leibler (KL) divergence instead of the IS divergence as a cost function for a practical reason [26]. Instead of Eq. (14), we assume

$$X_{mn} \sim \text{Poisson}(Y_{mn}), \quad (15)$$

where $\mathbb{E}[X_{mn}] = Y_{mn}$ holds. $\{X_{mn}\}_{m=1}^M$ and $\{Y_{mn}\}_{m=1}^M$ are the *amplitude* spectra of frame n because KL-NMF models are usually formulated in the amplitude domain by assuming the amplitude additivity [10, 18].

A nonparametric Bayesian extension called *infinite* CAR enables us to automatically estimate appropriate numbers of sources and filters according to the observation \mathbf{X} [26]. This technique is based on gamma process NMF [15].

Another extension of CAR is to force the amplitude spectrum of each source $\{S_{im}\}_{m=1}^M$ to have a harmonic structure [26]. If the source signal is a train of periodic impulses (an idealized model of the vocal chords), $\{S_{im}\}_{m=1}^M$ has a harmonic structure consisting of equally-spaced harmonic partials with the same weight. The optimal value of the F0 can be estimated such that the likelihood given by Eq. (15) is maximized. This technique of F0 estimation has a potential to solve the limitation of DAP.

3. PROPOSED MODEL

This section presents a nonparametric Bayesian approach called *infinite superimposed discrete all-pole* (iSDAP) modeling for source-filter decomposition of wavelet spectrograms. Our model can estimate multiple F0s at each frame and discover several kinds of instrument timbres (all-pole spectral envelopes) from polyphonic music audio signals. To achieve this, we integrate the technique of discrete all-pole (DAP) modeling [8] into the framework of composite autoregressive (CAR) modeling [17, 26] in a probabilistic manner. The iSDAP model can be regarded as a Bayesian extension of log-frequency source-filter NMF based on a single filter [19], and has all of the following features:

1. **Superimposed DAP modeling:** Our model can estimate the spectral envelope of each of harmonic structure contained in mixed sounds. The original DAP model can deal with only isolated sounds [8].
2. **Precise F0 modeling:** Each frame is allowed to contain a unique set of F0s (sources) for capturing fine

fluctuations of F0s (e.g., vibrato). The original CAR models [17, 26] assume that a common set of source spectra (semitone-level F0s) is shared over all frames.

3. **Log-frequency modeling:** Our source-filter model can deal with wavelet spectrograms that suit the characteristics of human audition by leveraging an advantage of DAP modeling that only discrete frequencies are required for spectral envelope estimation.
4. **Bayesian nonparametrics:** Our model can estimate effective numbers of sources and filters according to a given spectrogram by allowing unbounded (infinite in theory) numbers of sources and filters to be used.

3.1 Model Formulation

We explain a probabilistic model of iSDAP. Let \mathbf{X} be an $M \times N$ log-frequency amplitude spectrogram with M frequency bins and N frames. The nonnegative matrix \mathbf{X} is factorized in a similar way to Eq. (10) as follows:

$$X_{mn} \sim \text{Poisson} \left(\sum_{i=1}^{I \rightarrow \infty} \sum_{j=1}^{J \rightarrow \infty} \theta_{ni} \phi_j W_{nijm} H_{nij} \right), \quad (16)$$

where θ_{ni} is the local weight of source i at frame n , ϕ_j is the global weight of filter j , and H_{nij} is the gain of a pair of source i and filter j at frame n . $\{W_{nijm}\}_{m=1}^M$ is the amplitude spectrum derived from the source-filter pair at frame n . Note that θ_{ni} and W_{nijm} are allowed to vary over time to represent the F0 fluctuation unlike Eq. (10). We aim to perform sparse learning of weight vectors $\boldsymbol{\theta}_n = [\theta_{n1}, \dots, \theta_{nI}]^T$ and $\boldsymbol{\phi} = [\phi_1, \dots, \phi_J]^T$ when the number of sources I and the number of filters J go to infinity.

3.1.1 Parametric Functions

As shown in Figure 4, we force the amplitude spectrum $\{W_{nijm}\}_{m=1}^M$ to have a harmonic structure as follows:

$$W_{nijm} = \sum_{r=1}^R S_{mnir} F_{nijr}, \quad (17)$$

where R is the number of harmonic partials and $\{S_{mnir}\}_{m=1}^M$ is the monomodal spectrum of the r -th harmonic partial of source i at frame n given by

$$S_{mnir} = \exp \left(-\frac{1}{2\sigma^2} (f_m - (\mu_{ni} + 1200 \log_2 r))^2 \right), \quad (18)$$

where μ_{ni} is the F0 [cents] of source i at frame n , f_m is the log-frequency [cents] corresponding to the m -th bin, and σ^2 indicates energy diffusion around harmonic partials.

We then represent the weights of discrete harmonic partials, $\{F_{nijr}\}_{r=1}^R$, by using an all-pole transfer function in the log frequency domain as follows:

$$F_{nijr} = \frac{1}{\left| \sum_{p=0}^P a_{jp} e^{-\omega_{nir} p} \right|} = (\mathbf{a}_j^T \mathbf{U}(\omega_{nir}) \mathbf{a}_j)^{-\frac{1}{2}}, \quad (19)$$

where $\mathbf{a}_j \equiv [a_{j0}, \dots, a_{jP}]^T$, ω_{nir} is a normalized frequency [rad] corresponding to the r -th harmonic partial of source i at frame n , and $\mathbf{U}(\omega)$ is a $(P+1) \times (P+1)$ matrix with $[\mathbf{U}(\omega)]_{pq} = \cos(\omega(p-q))$. Note that F_{nijr} indicates the value of amplitude (not power). The Poisson likelihood

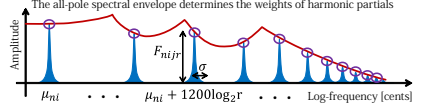


Figure 4. Composition of source i and filter j at frame n in the log-frequency domain.

(KL-NMF) is considered to fit the amplitude domain rather than the power domain [19].

3.1.2 Prior Distributions

We put gamma process (GaP) priors on infinite-dimensional vectors $\boldsymbol{\theta}_n$ and $\boldsymbol{\phi}$ as in [15, 26]. Specifically, we put independent gamma priors on elements of $\boldsymbol{\theta}_n$ and $\boldsymbol{\phi}$ as follows:

$$\theta_{ni} \sim \text{Gamma} \left(\frac{\alpha_\theta}{I}, \alpha_\theta \right), \quad \phi_j \sim \text{Gamma} \left(\frac{\alpha_\phi}{J}, \alpha_\phi \right), \quad (20)$$

where α_θ and α_ϕ are hyperparameters called concentration parameters. As J diverges to infinity, the vector $\boldsymbol{\phi}$ approximates an infinite vector drawn from a GaP with α_ϕ . It is proven that the effective number of filters, J^+ , such that $\phi_j > \epsilon$ for some number $\epsilon > 0$ is almost surely finite [15]. If J is sufficiently larger than α_ϕ (J is often called a truncation level in weak-limit approximation to infinite modeling), the GaP can be well approximated. The same reasoning can be applied to the GaP on $\boldsymbol{\theta}_n$. On the other hand, we put a standard Gamma prior on H_{nij} as follows:

$$H_{nij} \sim \text{Gamma}(a_H, b_H), \quad (21)$$

where a_H and b_H are shape and rate hyperparameters.

3.2 Variational Inference

The posterior over random variables $p(\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{H} | \mathbf{X}; \boldsymbol{\mu}, \mathbf{a})$ and parameters $\boldsymbol{\mu}$ and \mathbf{a} are determined such that a lower bound \mathcal{L} of the log-evidence $\log p(\mathbf{X}; \boldsymbol{\mu}, \mathbf{a})$ is maximized. Since this cannot be analytically computed, we use an approximate method called variational Bayes (VB), which restricts the posterior to a factorized form given by

$$q(\boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{H}) = \prod_{ni} q(\theta_{ni}) \prod_j q(\phi_j) \prod_{nij} q(H_{nij}). \quad (22)$$

Iteratively updating this posterior, we can monotonically increase a lower bound of the log-evidence given by

$$\begin{aligned} \log p(\mathbf{X}; \boldsymbol{\mu}, \mathbf{a}) &\geq \mathbb{E}[\log p(\mathbf{X} | \boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{H}; \boldsymbol{\mu}, \mathbf{a})] \\ &+ \mathbb{E}[\log p(\boldsymbol{\theta})] + \mathbb{E}[\log p(\boldsymbol{\phi})] + \mathbb{E}[\log p(\mathbf{H})] \\ &- \mathbb{E}[\log q(\boldsymbol{\theta})] - \mathbb{E}[\log q(\boldsymbol{\phi})] - \mathbb{E}[\log q(\mathbf{H})] \equiv \mathcal{L}_0, \end{aligned} \quad (23)$$

where the first term can be further lower bounded by Jensen's inequality on the concave logarithmic function as follows:

$$\begin{aligned} \mathbb{E}[\log p(\mathbf{X} | \boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{H}; \boldsymbol{\mu}, \mathbf{a})] &= \sum_{mn} X_{mn} \mathbb{E} \left[\log \sum_{ijr} \theta_{ni} \phi_j S_{mnir} F_{nijr} H_{nij} \right] \\ &- \sum_{mnijr} \mathbb{E} [\theta_{ni} \phi_j S_{mnir} F_{nijr} H_{nij}] + \text{const.} \\ &\geq \sum_{mnijr} \lambda_{mnijr} X_{mn} \mathbb{E} \left[\log \frac{\theta_{ni} \phi_j S_{mnir} F_{nijr} H_{nij}}{\lambda_{mnijr}} \right] \\ &- \sum_{mnijr} \mathbb{E} [\theta_{ni} \phi_j S_{mnir} F_{nijr} H_{nij}] + \text{const.} \end{aligned} \quad (24)$$

where $\lambda_{mni jr}$ is a normalized auxiliary variable such that $\sum_{i jr} \lambda_{mni jr} = 1$. The equality holds (*i.e.*, the lower bound of \mathcal{L}_0 is maximized) if and only if

$$\lambda_{mni jr} \propto \exp(\mathbb{E}[\log(\theta_{ni}\phi_j S_{mnir} F_{ni jr} H_{nij})]). \quad (25)$$

Using Eq. (24), the objective function of our model to be maximized, \mathcal{L} , is obtained as the lower bound of \mathcal{L}_0 . For convenience, we define $X_{mni jr}$ and $Y_{mni jr}$ as

$$X_{mni jr} = \lambda_{mni jr} X_{mn}, \quad (26)$$

$$Y_{mni jr} = \mathbb{E}[\theta_{ni}\phi_j S_{mnir} F_{ni jr} H_{nij}]. \quad (27)$$

3.3 Variational Bayesian Updating of θ , ϕ , and H

The VB updating rules are given by

$$q(\boldsymbol{\theta}) \propto \exp(\mathbb{E}_{q(\boldsymbol{\phi}, \mathbf{H})}[\log p(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{H}; \boldsymbol{\mu}, \boldsymbol{\alpha})]),$$

$$q(\boldsymbol{\phi}) \propto \exp(\mathbb{E}_{q(\boldsymbol{\theta}, \mathbf{H})}[\log p(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{H}; \boldsymbol{\mu}, \boldsymbol{\alpha})]), \quad (28)$$

$$q(\mathbf{H}) \propto \exp(\mathbb{E}_{q(\boldsymbol{\theta}, \boldsymbol{\phi})}[\log p(\mathbf{X}, \boldsymbol{\theta}, \boldsymbol{\phi}, \mathbf{H}; \boldsymbol{\mu}, \boldsymbol{\alpha})]).$$

The variational posterior of each random variable is set to be the same family as its prior distribution as follows:

$$\begin{aligned} q(\theta_{ni}) &= \text{Gamma}(a_{ni}^{\theta}, b_{ni}^{\theta}), & q(\phi_j) &= \text{Gamma}(a_j^{\phi}, b_j^{\phi}), \\ q(H_{nij}) &= \text{Gamma}(a_{nij}^H, b_{nij}^H). \end{aligned} \quad (29)$$

The variational parameters are given by

$$a_{ni}^{\theta} = \frac{\alpha_{\theta}}{I} + \sum_{mjr} X_{mni jr}, \quad b_{ni}^{\theta} = \alpha_{\theta} + \sum_{mjr} \mathbb{E}[\phi_j H_{nij}] W_{nijm},$$

$$a_j^{\phi} = \frac{\alpha_{\phi}}{J} + \sum_{mnr} X_{mni jr}, \quad b_j^{\phi} = \alpha_{\phi} + \sum_{mnr} \mathbb{E}[\theta_{ni} H_{nij}] W_{nijm},$$

$$a_{nij}^H = a_H + \sum_{mnr} X_{mni jr}, \quad b_{nij}^H = b_H + \sum_{mnr} \mathbb{E}[\theta_{ni}\phi_j] W_{nijm}.$$

To estimate the effective number of filters J^+ , we perform sparse learning. If $\mathbb{E}[\phi_j]$ becomes sufficiently small for some filter j , we degenerate it and $J \leftarrow J - 1$. A similar treatment is applied to $\mathbb{E}[\theta_{ni}]$. The proposed variational algorithm is gradually accelerated per iteration.

3.4 Multiplicative Updating of $\boldsymbol{\mu}$ and \boldsymbol{a}

To estimate parameters $\boldsymbol{\mu}$ and \boldsymbol{a} , we use the multiplicative update (MU) algorithm as in [1, 14]. In general, to optimize x , we represent the partial derivative of a ‘‘cost’’ function \mathcal{C} with respect to x as the difference of two positive terms, *i.e.*, $\frac{\partial \mathcal{C}}{\partial x} = R - R'$. An updating rule of x is then given by $x \leftarrow \frac{R}{R'} x$. Note that x becomes constant if the derivative is zero, and is updated in the opposite direction of the derivative. In this study the cost function is the negative lower bound of the log-evidence, $-L$.

First, we represent the partial derivative of $-L$ with respect to μ_{ni} as $\frac{\partial \mathcal{L}}{\partial \mu_{ni}} = R_{ni} - R'_{ni}$, where R_{ni} and R'_{ni} are positive terms given by

$$R_{ni} = \sum_{mjr} (\mu_{ni} + 1200 \log_2 r) X_{mni jr} + f_m Y_{mni jr}, \quad (30)$$

$$R'_{ni} = \sum_{mjr} f_m X_{mni jr} + (\mu_{ni} + 1200 \log_2 r) Y_{mni jr}. \quad (31)$$

The updating rule of μ_{ni} is given by

$$\mu_{ni} \leftarrow R_{ni}^{-1} R'_{ni} \mu_{ni}. \quad (32)$$

As in [1, 14], we then represent the partial derivative of $-L$ with respect to \boldsymbol{a}_j as $\frac{\partial \mathcal{L}}{\partial \boldsymbol{a}_j} = (\mathbf{R}_j - \mathbf{R}'_j) \boldsymbol{a}_j$, where \mathbf{R}_j and \mathbf{R}'_j are positive definite matrices given by

$$\mathbf{R}_j = \sum_{mnr} X_{mni jr} F_{ni jr}^2 \mathbf{U}(\omega_{nir}), \quad (33)$$

$$\mathbf{R}'_j = \sum_{mnr} Y_{mni jr} F_{ni jr}^2 \mathbf{U}(\omega_{nir}). \quad (34)$$

The updating rule of \boldsymbol{a}_j is given by

$$\boldsymbol{a}_j \leftarrow \mathbf{R}_j^{-1} \mathbf{R}'_j \boldsymbol{a}_j. \quad (35)$$

Finally, we forcibly adjust the scale of the filter $F_{ni jr}$ such that $\alpha_{j0} = 1$ for normalizing the filter. Although this step violates the convergence of the optimization algorithm, it was empirically found to work well.

3.5 Binary Piano-roll Estimation

To perform multipitch analysis, *i.e.*, make a binary piano-roll representation, we need to judge the existence of each semitone-level pitch at each frame. Using a trained model, we calculate an activation matrix $\mathbf{V} = \{V_{kn}\}_{k=1, n=1}^{88, N}$ over pitch k and frame n (continuous-valued piano-roll representation *e.g.*, the middle figure of Figure 5) by accumulating the expected amplitude of the first partial of source i , $\sum_j \mathbb{E}[\theta_{ni}\phi_j F_{ni jr} H_{nij}]$, into V_{kn} indicated by μ_{ni} . Finally, the activation matrix \mathbf{V} is normalized such that all the elements sum to unity, *i.e.*, $\sum_{kn} V_{kn} = 1$.

There are several approaches to binary piano-roll estimation. The common approach is to make a binary decision based on a threshold η . Another approach is to define a hidden Markov model (HMM) and use the Viterbi-search algorithm for estimating a sequence of hidden binary states $\{Z_{kn}\}_{n=1}^N$ from a sequence of pitch-existence likelihoods $\{V_{kn}^p\}_{n=1}^N$ for each pitch k , where p controls the dynamic range. In our implementation, $p = 0.2$ and the transition matrix is $[0.8, 0.2; 0.01, 0.99]$ in the Matlab notation.

4. EVALUATION

We report comparative experiments that were conducted for evaluating the performance of the ISDAP model in multipitch analysis of piano music. Since the proposed model assumes that input mixture signals contain only harmonic sounds, we also tested the use of harmonic and percussive source separation (HPSS) [12] as a preprocessor.

4.1 Experimental Conditions

We used 30 pieces (labeled as ‘‘ENSTDkCl’’) selected from the MAPS database [9] that contain stereo signals sampled at 44.1 [kHz]. The audio signals were converted to monaural signals and truncated to 30 [s] from the beginning as in [2, 4, 21, 22, 24]. The amplitude spectrogram of each piece over the frequency bins ranging from 0 [cents] (16.325 [Hz]) to 12000 [cents] (16717 [Hz]) was obtained by performing the wavelet transform with a Gabor wavelet, a frequency interval of 10 [cents], and a shifting interval of 10 [ms], *i.e.*, $M = 1200$ and $N = 3000$. The other quantities were $I = 88$, $J = 3$, $R = 20$, $P = 13$, and $\sigma = 25$. The priors were set to be less informative, *i.e.*,

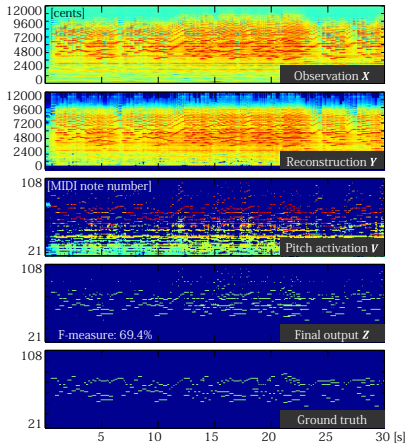


Figure 5. Analysis of MUS-mz_333_3.ENSTDkCl.

$\alpha_\phi = \alpha_\phi = a_H = 1$, and $b_H = \mathbb{E}_{\text{emp}}[X_{mn}]^{-1}$. Since \mathbf{X} contained only piano sounds, the truncation level $J = 3$ worked well (two filters were degenerated in this experiment, *i.e.*, $J^+ = 1$). The values of $\{\mu_{ni}\}_{i=1}^88$ were initialized as the frequencies corresponding to the 88 keys ranging from 900 [cents] to 9600 [cents]. The value of each α_{jp} ($1 \leq p \leq P$) was drawn from a Gaussian with a zero mean and a small variance of 0.01. The variational posteriors were initialized as the corresponding priors.

The proposed model was tested under possible combinations of preprocessing (with or without HPSS) and post-processing (thresholding or Viterbi decoding). HPSS was performed in the log-frequency domain. The model with a single filter ($J = J^+ = 1$) was also tested in a supervised setting. A set of filter coefficients \mathbf{a}_1 was pretrained from 264 isolated sounds of the same or different piano (ENSTDkCl in a closed test or SptkBGCl in an open test) by using LPC, and kept constant during multipitch analysis.

The estimation results were evaluated in terms of the frame-level recall/precision rates and F-measure as in [24]:

$$\mathcal{R} = \frac{\sum_n c_n}{\sum_n r_n}, \quad \mathcal{P} = \frac{\sum_n c_n}{\sum_n e_n}, \quad \mathcal{F} = \frac{2\mathcal{R}\mathcal{P}}{\mathcal{R} + \mathcal{P}}, \quad (36)$$

where r_n , e_n , and c_n are the numbers of ground truth, estimated and correct pitches on frame n , respectively. The threshold η was determined as $\eta = 10^{-1.3}$ without HPSS and $\eta = 10^{-1.5}$ with HPSS.

4.2 Experimental Results

The experimental results shown in Figure 5 and Table 1 indicate the great potential of the iSDAP model. The model supervised in the open condition (67.3%) significantly outperformed the iCAR model formulated in the linear frequency domain (48.4%) [26] and tied with the state-of-the-art methods, *e.g.*, harmonic NMF (67.7%) [24], NMF with group sparsity (71.3%) [21], and NMF with Hellinger

Filter learning	HPSS	HMM	\mathcal{R}	\mathcal{P}	\mathcal{F}
Unsupervised			55.3	57.9	56.6
	✓	✓	62.2	60.2	61.2
	✓	✓	62.4	64.3	63.4
Supervised (open test)	✓		62.4	67.0	64.4
	✓	✓	69.9	64.5	67.3
Supervised (close test)	✓		59.4	69.1	63.9
	✓	✓	67.4	67.8	67.6

Table 1. Experimental results of multipitch analysis for 30 piano pieces labeled as ENSTDkCl.

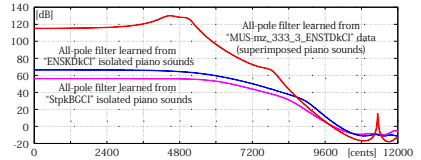


Figure 6. All-pole filters learned from isolated sounds and a piano piece (mixed sounds) in the log-frequency domain.

space coding (66.5%) [22]). While many recent methods need to pretrain a dictionary of basis spectra for reasonable decomposition [2, 4, 21, 24], our model works well (65.8%) even in the completely unsupervised condition. As shown in Figure 6, a filter learned from a music signal dropped faster than the pretrained filters because the model failed to capture higher-order overtones even in the log-frequency domain due to the strong inharmonicity of piano sounds. Nonetheless, the learned filter acted as an effective constraint on the relative weights of harmonic partials.

There would be much room for improving the performance. KL-NMF [18] and IS-NMF [10] are special cases of β -divergence NMF [11, 20] with $\beta = 1, 0$, respectively. It was reported that the use of an intermediate divergence with $\beta = 0.5$ significantly improves the performance by about 5% [24]. Similar findings were reported in the context of source separation [13]. This calls for the use of the Tweedie likelihood instead of the Poisson likelihood [6].

5. CONCLUSION

We presented a new nonparametric Bayesian approach to source-filter NMF called infinite superimposed discrete all-pole (iSDAP) modeling that can decompose a *wavelet* spectrogram into three kinds of factors, *i.e.*, harmonic sources, all-pole filters, and time-varying gains of source-filter pairs. Our model clearly outperformed its counterpart called the iCAR model formulated in the linear frequency domain. One important research direction is to build a unified model of harmonic and percussive sounds. To bridge the gap between multipitch analysis and music transcription, we plan to incorporate a prior distribution on the time-frequency positions of musical notes into a Bayesian framework.

Acknowledgment: This study was partially supported by JST OngaCREST Project, JSPS KAKENHI 24220006, 26700020, and 26280089, and Kayamori Foundation.

6. REFERENCES

- [1] R. Badeau and A. Ozerov. Multiplicative updates for modeling mixtures of non-stationary signals in the time-frequency domain. In *European Signal Processing Conference (EUSIPCO)*, 2013.
- [2] E. Benetos, R. Badeau, T. Weyde, and G. Richard. Template adaptation for improving automatic music transcription. In *International Society for Music Information Retrieval Conf. (ISMIR)*, pages 175–180, 2014.
- [3] N. J. Bryan, G. Mysore, and G. Wang. Source separation of polyphonic music with interactive user-feedback on a piano roll display. In *International Society for Music Information Retrieval Conference (ISMIR)*, pages 119–124, 2013.
- [4] J. J. Carabias-Orti, T. Virtanen, P. Vera-Candeas, N. Ruiz-Reyes, and F. J. Cañadas-Quesada. Musical instrument sound multi-excitation model for non-negative spectrogram factorization. *IEEE Journal of Selected Topics in Signal Proc.*, 5(6):1144–1158, 2011.
- [5] A. T. Cemgil. Bayesian inference for nonnegative matrix factorisation models. *Computational Intelligence and Neuroscience*, 2009:Article ID 785152, 2009.
- [6] U. Şimşekli, A. Cemgil, and Y. K. Yılmaz. Learning the β -divergence in Tweedie compound Poisson matrix factorization models. In *International Conference on Machine Learning (ICML)*, pages 1409–1417, 2013.
- [7] J.-L. Durrieu, G. Richard, B. David, and C. Févotte. Source/filter model for unsupervised main melody extraction from polyphonic audio signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):564–575, 2010.
- [8] A. El-Jaroudi and J. Makhoul. Discrete all-pole modeling. *IEEE Transactions on Signal Processing*, 39(2):411–423, 1991.
- [9] V. Emiya, R. Badeau, and B. David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Transactions on Audio, Speech and Language Processing*, 18(6):1643–1654, 2010.
- [10] C. Févotte, N. Bertin, and J.-L. Durrieu. Nonnegative matrix factorization with the Itakura-Saito divergence: With application to music analysis. *Neural Computation*, 21(3):793–830, 2009.
- [11] C. Févotte and J. Idier. Algorithms for nonnegative matrix factorization with the beta-divergence. *Neural Computation*, 23(9):2421–2456, 2011.
- [12] D. FitzGerald. Harmonic/percussive separation using median filtering. In *International Conference on Digital Audio Effects (DAFx)*, 2010.
- [13] D. FitzGerald, M. Cranitch, and E. Coyle. On the use of the beta divergence for musical source separation. In *Irish Signals and Systems Conf.*, pages 1–6, 2008.
- [14] R. Hennequin, R. Badeau, and B. David. NMF with time-frequency activations to model nonstationary audio events. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):744–753, 2011.
- [15] M. Hoffman, D. Blei, and P. Cook. Bayesian nonparametric matrix factorization for recorded music. In *International Conference on Machine Learning (ICML)*, pages 439–446, 2010.
- [16] F. Itakura and S. Saito. Analysis synthesis telephony based on the maximum likelihood method. In *International Congress on Acoustics (ICA)*, pages C17–C20, 1968.
- [17] H. Kameoka and K. Kashino. Composite autoregressive system for sparse source-filter representation of speech. In *IEEE International Symposium on Circuits and Systems (ISCAS)*, pages 2477–2480, 2009.
- [18] D. Lee and H. Seung. Algorithms for non-negative matrix factorization. In *Neural Information Processing Systems (NIPS)*, pages 556–562, 2000.
- [19] T. Nakamura, K. Shikata, N. Takamune, and H. Kameoka. Harmonic-temporal factor decomposition incorporating music prior information for informed monaural source separation. In *International Society for Music Information Retrieval Conference (ISMIR)*, pages 623–628, 2014.
- [20] M. Nakano, H. Kameoka, J. Le Roux, Y. Kitano, N. Ono, and S. Sagayama. Convergence-guaranteed multiplicative algorithms for non-negative matrix factorization with beta divergence. In *International Workshop on Machine Learning for Signal Processing (MLSP)*, pages 283–288, 2010.
- [21] K. O’Hanlon and M. D. Plumbley. Polyphonic piano transcription using non-negative matrix factorisation with group sparsity. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2214–2218, 2014.
- [22] K. O’Hanlon, M. Sandler, and M. D. Plumbley. Matrix factorisation incorporating greedy Hellinger sparse coding applied to polyphonic music transcription. In *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 3112–3116, 2015.
- [23] P. Smaragdīs, C. Févotte, G. Mysore, N. Mohammadī, and M. Hoffman. Dynamic source separation using nonnegative factorizations: A unified view. *IEEE Signal Processing Magazine*, 31(3):66–75, 2014.
- [24] E. Vincent, N. Bertin, and R. Badeau. Adaptive harmonic spectral decomposition for multiple pitch estimation. *IEEE Transactions on Audio, Speech and Language Processing*, 18(3):528–537, 2010.
- [25] T. Virtanen and A. Klapuri. Analysis of polyphonic audio using source-filter model and non-negative matrix factorization. In *NIPS Workshop on Advances in Models for Acoustic Processing*, 2009.
- [26] K. Yoshi and M. Goto. Infinite composite autoregressive models for music signal analysis. In *International Society for Music Information Retrieval Conference (ISMIR)*, pages 79–84, 2012.

MELODIC SIMILARITY IN TRADITIONAL FRENCH-CANADIAN INSTRUMENTAL DANCE TUNES

Laura Risk

Lillio Mok

Andrew Hankinson

Julie Cumming

Schulich School of Music
Centre for Interdisciplinary Research in Music Media and Technology (CIRMMT)
McGill University

laura.risk@mail.mcgill.ca

ABSTRACT

Commercial recordings of French-Canadian instrumental dance tunes represent a varied and complex corpus of study. This was a primarily aural tradition, transmitted from performer to performer with few notated sources until the late 20th century. Practitioners routinely combined tune segments to create new tunes and personalized settings of existing tunes. This has resulted in a corpus that exhibits an extreme amount of variation, even among tunes with the same name. In addition, the same tune or tune segment may appear under several different names.

Previous attempts at building systems for automated retrieval and ranking of instrumental dance tunes perform well for near-exact matching of tunes, but do not work as well in retrieving and ranking, in order of most to least similar, variants of a tune; especially those with variations as extreme as this particular corpus. In this paper we will describe a new approach capable of ranked retrieval of variant tunes, and demonstrate its effectiveness on a transcribed corpus of incipits.

1. INTRODUCTION

Commercial recordings of French-Canadian instrumental dance tunes from the 1920s through the 1980s document a working-class repertoire now celebrated as the traditional instrumental music of Québec [19]. However, the musical contents of these recordings remain largely unexamined. The only detailed musicological study is limited to a subset of metrically irregular tunes [8]. In this paper we outline the challenges associated with studying this repertoire and describe a new system developed to aid in finding and ranking similarity between tunes in this repertoire. This system was built in response to two musicological challenges: to determine the degree of shared repertoire among early commercial recording artists in Montréal, and to identify how a single tune is varied in different renditions. Using our system we have identified a number of concordant tunes

(versions of the same tune) previously unrecognized as being musically related.

Broadly speaking, the traditional instrumental music of Québec is similar to the instrumental traditions of Ireland, Scotland and the United States. The repertoire consists primarily of short, fast-paced dance tunes usually performed on the violin, accordion, or harmonica. With very few exceptions, each tune has at least two strains (sections), commonly labeled “A” and “B.” Many of the tunes have their roots in British Isles and American fiddling traditions, though others are derived from popular French songs or early twentieth-century marching-band repertoire [8; 13; 23].

2. THE CHALLENGE

The French-Canadian tradition has developed almost exclusively as an aural and recorded tradition. With few notated sources, musicians would often learn tunes “on the fly,” constructing their own versions from memory and injecting their own personal style. From the 1930s through the 1960s radio broadcasts played a significant role in aural transmission of this repertoire. One musician recalled, “I would listen to the radio with my brother, and afterwards we would sing the melodies in our room. We spent our time constantly asking ourselves if it was really correct” [9]. As a result of this mode of transmission, and in the absence of a culture of “correctness” [7], many tunes performed and recorded in Québec exist in multiple, equally valid settings. Tunes would be modified by transposing all or part of the tune, reworking melodic figurations, adding and subtracting beats, composing new strains, or combining strains from several tunes to form new tunes. This diversity of interpretation is clearly documented on commercial recordings from the era.

Tune titles were often lost or altered in transmission. Fiddler Yvon Mimeault, for instance, assigned his own titles to most of the tunes that he learned from the radio in the 1950s [14]. Other musicians renamed tunes quite intentionally. In the 1920s and 30s, fiddler Isidore Soucy sometimes recorded a tune for one record label under one title, and within months recorded the same tune for a different record label under a different title. The tune “Money Musk” is an exception. Between 1920 and 1980 it was recorded over twenty times, frequently with significant melodic and rhythmic variation and



© Laura Risk, Lillio Mok, Andrew Hankinson, Julie Cumming. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Laura Risk, Lillio Mok, Andrew Hankinson, Julie Cumming. “Melodic Similarity in Traditional French-Canadian Instrumental Dance Tunes,” 16th International Society for Music Information Retrieval Conference, 2015.

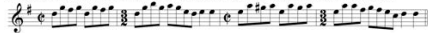
additional strains, but almost always under the same title.

To illustrate with one example, fiddlers Isidore Soucy, Joseph Ovila LaMadeleine, and Joseph Allard all released settings of the same melody in 1928. Their recordings were titled, respectively, “Reel du bon vieux temps,” “Reel princesse,” and “Reel de Mme. Renault” (note that Soucy reverses the order of the A and B strains with respect to LaMadeleine and Allard). Although these are recognizably the same melody, they have a significant degree of melodic and metrical variation (figure 1). Audio files for all three are available through the Virtual Gramophone website of Library and Archives Canada [20].

Joseph Allard, “Reel de Mme. Renault” (B strain), Victor 263531-B, 1928.



Joseph Ovila LaMadeleine, “Reel princesse” (B strain), Starr 15394-B, 1928.



Isidore Soucy, “Reel du bon vieux temps” (A strain), Starr 15406-A, 1928.

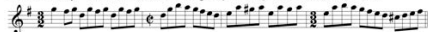


Figure 1: Incipits for three variants of a strain recorded in 1928.

The earliest recording sessions of traditional instrumental music in Québec were quick and largely unrehearsed. A pianist or guitarist would usually accompany a soloist with minimal rehearsal time prior to recording. Some of the recordings also include accompaniment on jaw harp (*gumbarde*) or spoons. The performances are unedited and were usually completed on either the first or second take. Some contain obvious musical errors, such as missed entries or wrong notes.

These performances were pressed to 78 RPM records, with one three-minute rendition of a tune per side [25]. These recordings contain a significant amount of background noise introduced in the recording chain, along with the standard problems of the 78 RPM format such as hiss, pops, and clicks.

The noisy recording environment and the relatively poor quality of the recordings result in recordings that are difficult to follow, even for human listeners. Due to these difficulties we decided not to explore signal-based approaches to analyzing this repertoire. Instead, our approach was to: 1) transcribe the A and B strains of a recording into MusicXML using a notation editor (Finale), and 2) devise a system for analyzing and computing the distance between two variants of the same tune.

Duval [8] estimates that the traditional instrumental music of Québec contains at least 5000 distinct tunes, not including variants. This gives a potential corpus of well over 10,000 strains. We are currently using our system to parse a database containing 710 strains. Of these, 667 were recorded between 1923 and 1929 and 59 strains are from renditions of “Money Musk” (16 strains of “Money Musk” were recorded between 1923 and 1929). This collection contains approximately 85% of all French-Canadian recordings of traditional instrumental

music on the violin prior to 1930, and approximately 50% of those recorded on any instrument prior to 1930. This selection of repertoire is clearly not random, but rather reflects the imperatives of several musicological questions, as discussed below.

3. PREVIOUS WORK

Scholars of aural traditions have long been fascinated by repertoire variation, and comparative studies abound. Bayard [2] proposed an influential theory of “tune families” by which the bulk of British Isles and North American folk song melodies could be categorized as variants of a small number of distinct prototypical melodies. Cowdery [4], drawing examples from Irish traditional music, pointed out that musicians do not think in terms of abstract prototypes but rather create new tunes or variants by reworking and combining segments of known repertoire. He argued that tune families were more appropriately defined by the presence of recurring melodic motives, and not by their degree of deviation from a “standard” or “ideal” version of the tune.

Our query and ranking system is intended to help scholars study the diversity of melodic variants within a given corpus. Musically, our approach is similar to the approach described in three previous studies. Ó Súilleabháin [22] analyzed the melodic variations of Irish fiddler Tommie Potts according to a framework of “set accented tones.” Goertzen [10] argued that seemingly disparate variants of Texas contest-style fiddle tunes are linked to a shared sense of each tune’s “essence,” itself composed of tune-specific musical markers. Duval [8] analyzed temporal variation as an innovative element of performance practice in French-Canadian tunes. However, none of these studies were performed using computational tools.

Several existing online resources allow users to search fiddle tunes by musical incipit. Both the Scottish Music Index [12] and the Traditional Tune Archive [18] classify tunes according to numerical theme codes that contain the scale degrees of the strong beats of the first two bars. Users may search on these sites for theme codes that exactly match a given string and that begin with that string, but may not search for tune variants.

TunePal [6] is a tool that translates audio into symbolic notation (ABC) and then compares that notation to a crowd-sourced database of traditional Irish tunes using an edit-distance function. TunePal looks for exact or similar strings of ABC regardless of metrical placement and is effective at identifying tunes and tune settings with a small amount of melodic variation (provided there has been no transposition). However, variants with significant melodic variation may not be considered a match.

Van Kranenburg [26] presents a comprehensive survey of computational modelling of similarity to Dutch folk songs. He concludes that identifying characteristic motifs is the most important factor when determining similarity between two melodies. As well,

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
w	[nan]	[0.5]	[2.5]	[4.0]	[2.5]	[-1.0]	[-0.5]	[-1.0]	[2.5]	[0.5]	[2.5]	[0.5]	[-2.0]	[-1.5]	[-0.5]	[-1.0]	[nan]
0.0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1.0	-0.5	-2	-1.5	1.5	4.5	-2	-1.5	0	2	-2	2	-2	2.5	-1	-1.5	1.5	NaN
2.0	-2.5	-3.5	0	0	2.5	-3.5	-1.5	2	0	0	0	0.5	1.5	-2.5	0	NaN	NaN
3.0	-4	-2	4.5	4	1	-3.5	0.5	0	2	-2	2.5	-0.5	0	-1	NaN	NaN	NaN
4.0	-2.5	2.5	2.5	2.5	1	-1.5	-1.5	2	0	0.5	1.5	-2	1.5	NaN	NaN	NaN	NaN
5.0	2	0.5	1	2.5	3	-3.5	0.5	0	2.5	-0.5	0	-0.5	NaN	NaN	NaN	NaN	NaN
6.0	0	-1	1	4.5	1	-1.5	-1.5	2.5	1.5	-2	1.5	NaN	NaN	NaN	NaN	NaN	NaN
7.0	-1.5	-1	3	2.5	3	-3.5	1	1.5	0	-0.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN
8.0	-1.5	1	4.5	1	-1	0	0	1.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
9.0	0.5	-1	3	2.5	3.5	-2	-1.5	1.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
10.0	-1.5	1	1	5	2.5	-3.5	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
11.0	0.5	-1	3.5	4	1	-2	-1.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
12.0	-1.5	1.5	2.5	2.5	2.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
13.0	1	0.5	1	4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
14.0	0	-1	2.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
15.0	-1.5	0.5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
16.0	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figure 2: Interval matrix for the B strain of “Reel de Mme. Renault” (Joseph Allard, 1928; see figure 1). The column headings (1 to 17) indicate the strong beats. The row headings (0.0 to 16.0) indicate the metric offset from a given strong beat. The first row (w) gives the weak beats for each strong beat as intervals from that strong beat. Interval values indicate number of whole steps.

he demonstrates that the frequency of global features (interval features and other pitch-based features, duration-ratio features, and other rhythmic features) is not sufficient to identify similar melodies, but that searching for ordered sequences of certain features at a local level may help locate similar song melodies. The system described in this paper builds on these findings.

4. METHODOLOGY

The design of our system exploits some general characteristics of French-Canadian instrumental dance tunes. We determine an optimal alignment between two four-measure incipits and then compare certain features at corresponding locations in the incipits. Transposition is common in this repertoire and tonal center is sometimes ambiguous, so all tunes are internally represented as a melodic contour. Each tune has two or more strains that function more or less independently and must therefore be treated separately. Most strains may be uniquely identified by a four-bar incipit. Since most tunes repeat after four measures, a transcribed incipit contains the primary motivic material for that strain. As in related British Isles and North American fiddle repertoires, metrical placement matters: the notes that fall on the strong beats are more essential to the identity of a strain than those on the weak beats [1].

Our system requires a set of strains in symbolic format and operates in two phases. A construction phase is used to build matrix representations of each strain’s incipit, after which a comparison phase computes the pairwise similarity between matrices. The construction phase first scans and truncates the prepared strains to a four-measure incipit by using the `music21` toolkit [5]. Melodic, or horizontal, intervals between the first note on the first strong beat and the notes on all other strong beats are then identified using the VIS analysis framework [1]. Horizontal intervals between each note on a weak beat and the note on its preceding strong beat are also indexed and stored. The resulting feature vectors of these pair-wise indexed intervals create an interval matrix. Each *j*th column of the matrix represents the horizontal intervals of the melody between the *j*th strong

beat and all subsequent strong beats. Figure 2 shows the interval matrix for the first strain shown in figure 1.

The comparison phase aligns the strong beats of two interval matrices before computing feature similarity. For matrices of the same length, we follow earlier musicological studies of related British Isles traditions [11; 22] and align the strong beats of the incipits bijectively (beat-to-beat). In other words, the idiomatic alignment for two equal-length incipits *A* and *B* is the *i*th beat in *A* to the *i*th beat in *B*. For matrices of different lengths, a standard longest-common-subsequence (LCS) dynamic programming algorithm [17; 21] is used to find the best alignment between two incipits considering the note on each strong beat.

After alignment this phase iteratively matches the notes of each aligned strong and weak beat in the two matrices. Non-matching notes are checked for three possible musical variation techniques: displacement, reversal and contour similarity. The aligned pair of strong beats (*i, j*) is non-matching when the note on beat *i* in *A* is not the same as the note on beat *j* in *B*. Thus, given two incipits *A* and *B* and each *k*th pair (*i, j*) of aligned but non-matching strong beats, *displacement* of notes on strong beats to a corresponding weak beat occurs when the note on the *i*th beat is amongst the notes in the weak beats after the *j*th beat, or vice versa. A *reversal* of notes on strong beats occurs when the note on the *i*th beat is the same as that on beat *j + 1*, and vice versa. Finally, *contour similarity* is detected when the horizontal interval between the notes on beats *i* and *i + 1* are the same as the interval between beats *j* and *j + 1*.

To account for cases where the incipits may otherwise match but have different notes on the first aligned strong beat, the algorithm moves to the next column of both interval matrices and repeats the comparison phase. This recursive strategy is only used on up to half the interval columns of the incipits. For each repetition of the comparison phase, a similarity matrix is constructed from the results of analyzing the matching, displacement, reversal, and contour similarity of the aligned strong beats in the corresponding interval columns between incipit pairs.

Each row of a similarity matrix corresponds to the

	Best Result:					
	1	2	3	4	5	6
Strong Beat Comparison	[0.0, 0, 0]	[-2.0, 1, 1]	[nan, 2, 2]	[-2.0, 3, 3]	[2.5, 4, 4]	[nan, 5, 5]
Displacement Comparison (Strong-Weak)	NaN	NaN	0	NaN	NaN	1
Weak Beats Comparison (Matched Strongs)	[1.0, 1.0]	[1.0, 1.0]	NaN	[1.0, 1.0]	[0, 0]	NaN
Weak Beats Comparison (Mismatched Strongs)	NaN	NaN	[1.0, 1.0]	NaN	NaN	[0, 0]
Contour Comparison (Strongs)	NaN	NaN	NaN	NaN	NaN	NaN
Contour Comparison (Weeks)	NaN	NaN	NaN	NaN	NaN	NaN
Reversal Comparison (Strongs)	NaN	NaN	NaN	NaN	NaN	NaN
Reversal Comparison (Weeks)	NaN	NaN	NaN	NaN	NaN	NaN
Shorter Incipit Length	17	17	17	17	17	17
Longer Incipit Length	17	17	17	17	17	17
Number of Truncations	1	1	1	1	1	1
Best Similarity Measure: 63.1067961165						

Figure 3: The similarity matrix comparing the B strain of “Reel de Mme. Renault” (Joseph Allard, 1928) and the B strain of “Reel princesse” (Joseph Ovila LaMadeleine, 1928). See figure 1 for incipits. This figure shows the feature values for the first six matched strong beats in the two incipits.

result of a particular feature analysis, and the columns represent the k th pair of aligned strong beats from each incipit’s interval matrix. Each cell in the similarity matrix is thus the result of a feature analysis for a given strong beat pair between incipits A and B . Entries in each similarity matrix are then combined with a weighting factor to yield a single similarity measure for each similarity matrix (figure 3). The “Strong Beat Comparison” matrices indicate the value of matching strong beats and identify aligned strong beats. The “Weak Beats Comparison” matrices indicate the fraction, and relative order, of matching weak beats.

Our weighting scheme is determined by trial and error. Up to 85% of the weight value is assigned to matching strong beats, displaced or reversed strong beats, and matching contour, with the remainder used for weighting matching weak beats. This weighting scheme has the effect of selecting incipits with a high percentage of matching strong beats and then ranking those selected according to number of matching weak beats.

5. RESULTS

Two outputs from this system may be useful to musicians, musicologists, and other researchers: the similarity matrices, which allow users to directly compare two incipits, and a ranked list of similarity measures between one strain and all other strains in the database.

We attempted a traditional precision and recall analysis but found it to be an unsuitable measure of effectiveness. It returned unnaturally high results because the weighting was determined to maximize precision and recall for known concordant strains. As noted in the discussion below, precision and recall for 25 variants of the A strain of “Money Musk” were either 100% ($n=10$) or 96% ($n=15$), given 24 relevant items, 24 retrieved items, and a database of 59 items.

Figure 4 gives incipits for the top 5 strains in the ranked list for the B strain of Joseph Allard’s “Reel de Mme. Renault,” as compared to 666 other strains recorded between 1923 and 1929. A human-provided musical analysis identified strains 1 and 3 (“Reel du bon vieux temps,” “Reel princesse”) as the only concordances in the database.

QUERY:

Joseph Allard, “Reel de Mme. Renault” (B strain), Victor 26351-B, 1928.



RESULTS:

Isidore Soucy, “Reel du bon vieux temps” (A strain), Starr 15406-A, 1928.

Rank 1, Similarity measure 74.9



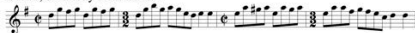
Isidore Soucy, “Money Musk” (B’ strain), Starr 15302-B, 1927.

Rank 2, Similarity measure 63.6



Joseph Ovila LaMadeleine, “Reel princesse” (B strain), Starr 15394-B, 1928.

Rank 3, Similarity measure 63.1



Joseph Allard, “Marche Sir Wilfrid Laurier” (C strain), Victor 263578-A, 1929.

Rank 4, Similarity measure 57.8



Arthur-Joseph Boulay, “Reel du Cowboy” (B strain), Victor 263582-A, 1929.

Rank 5, Similarity measure 57.3



Figure 4: Similarity results for a query of the B strain of “Reel de Mme. Renault” (Joseph Allard, 1928)

To test our system, we randomly selected a test set of 100 strains from the full database of 710 strains. We then selected query strains at random from within this test set. We confirmed via a human-supplied musical analysis that each query strain had at least one concordance in the test set. Those that did not were discarded and new strains were randomly selected, until we reached a total of 10. Approximately 50% of the strains in the full database are not concordant with any other strains in the database.

We compared each of these query strains to the test set using four different ranking approaches (figure 5). The Levenshtein and Geometric Distance measures were drawn from the similarity evaluation system described in [16]. The MATT2 system [6] is designed for a repertoire of Irish tunes and is the search algorithm underlying the TunePal app described earlier. It presumes that transcriptions of tunes on “unusually pitched” instruments (instruments with repertoire-specific and non-standard tunings) have been normalized to a single fundamental pitch (“transposition invariance”).

QUERY: Performer, year, tune title (strain); number of concordant strains (CS) in test set.	* How many concordant strains (CS) are in the top 10 results for the query strain? * What are the rankings of those CS? Note: CS were determined by human-provided analysis			
	Our system	MATT2 (fundamental pitch normalized)	Levenshtein Distance	Geometric Distance
Allard, 1928, "Reel du Pendu" (B); 2 CS	2 CS; rank 1, 2; SM 77, 69.	2 CS; rank 1, 2.	0 CS.	0 CS.
Allard, 1929, "Reel des Violonneux" (B); 2 CS	2 CS; rank 1, 2; SM 89, 75.	2 CS; rank 1, 3.	2 CS; rank 1, 2.	2 CS; rank 1, 5.
Boulay, 1923, "Gigues Pot-Pourri" (2 nd tune, A); 4 CS	4 CS; rank 1, 2, 3, 4; SM 91, 91, 90, 80.	4 CS; rank 1, 2, 3, 4.	0 CS.	1 CS; rank 6.
Duchesne, 1938, "Money Musk American" (A); 4 CS	4 CS; rank 1, 2, 3, 4; SM 93, 91, 90, 75.	4 CS; rank 1, 2, 3, 4.	0 CS.	1 CS; rank 2.
Joyal, 1956, "Money Musk" (B); 3 CS	3 CS; rank 1, 2, 7; SM 80, 76, 46.	2 CS; rank 2, 3.	1 CS; rank 1.	1 CS; rank 1.
Lajoie, 1951, "Money Musk" (B); 3 CS	3 CS; rank 1, 2, 3; SM 80, 67, 55.	2 CS; rank 1, 2.	0 CS.	0 CS.
LaMadeleine, 1927, "Quadrille Franco-Américain 6e partie" (A); 1 CS	1 CS; rank 1; SM 100.	1 CS; rank 1.	1 CS; rank 1.	1 CS; rank 1.
Potvin, 1980, "Money Musk" (A); 4 CS	4 CS; rank 1, 2, 3, 4; SM 93, 91, 86, 75.	4 CS; rank 1, 2, 3, 4.	2 CS; rank 4, 6.	0 CS.
Soucy, 1925, "Gigues irlandaises no. 2" (2 nd tune, A); 2 CS	2 CS; rank 1, 2; SM 91, 49.	2 CS; rank 1, 2.	1 CS; rank 1.	2 CS; rank 1, 2.
Soucy, 1927, "Quadrille Laurier, 3e partie" (A); 1 CS	1 CS; rank 1; SM 60.	0 CS.	1 CS; rank 1.	1 CS; rank 1.

Figure 5: Query results for 10 strains out of a test set of 100 strains.

As expected, all of these analytical systems performed well when identifying exact or near-exact matches. However, our system was able to identify more extreme melodic and metrical variants and to supply a ranking of those variants. Our system also identified concordances in transposed keys.

6. DISCUSSION

We built this system in response to two musicological challenges. First, we wanted to identify concordances in the earliest commercial recordings of French-Canadian tunes in order to determine the degree of shared repertoire among these recording artists. Second, we wanted to analyze variation technique in a single tune, "Money Musk," due to its popularity in recordings of the era.

By applying our approach to a database of 667 strains recorded between 1923 and 1929, we were able to identify nearly 150 concordant strains. Most of these concordances were previously unrecognized as related tunes.

Using these results in combination with archival research, we have been able to identify patterns of musical borrowing for certain musicians. Of the 16 sides that fiddler Isidore Soucy recorded for Columbia Records in New York City in 1927–1929 [24], for instance, eight were tunes that he had recorded for the Starr label in Montréal only a few months earlier, most

under different titles. In contrast, when Soucy borrowed from his Starr releases for other Starr recordings, he usually re-recorded only single strains (combined with new material or with strains borrowed from other tunes).

We have been able to document the musical links between a small group of fiddlers living and working in the Montréal region in the late 1920s. These musicians often re-recorded the same tunes and strains within weeks of each other. Joseph Allard and Isidore Soucy, for example, recorded the same tune in late 1928 under the titles "Quadrille Acadien" and "Gigue Indienne," respectively (Victor 263543–A, Starr 15517–A). In the summer of 1927, Willie Ringuette and Isidore Soucy added the same C strain to two different tunes (Starr 15347–A, Starr 15363–B).

Finally, we were able to identify French-Canadian variants of many common North American tunes such as "Soldier's Joy," "Haste to the Wedding," "Fisher's Hornpipe," "Bristol Hornpipe," "Rickett's Hornpipe," "Chicken Reel," "Irish Washerwoman," "Keel Row," "Lord McDonald" and "Home Sweet Home."

We applied our system to a database of 59 strains drawn from 13 renditions of "Money Musk." All "Money Musk" settings include some version of two particular strains, usually labeled A and B, though not all performers play the A strain first and the B strain second. These strains may be easily recognized: both begin with a down-and-back motion that outlines a tonic chord, though the A strain begins on the fifth scale degree and the B on the first scale degree. (For three early and quite varied renditions of "Money Musk," listen to recordings on the Virtual Gramophone by Isidore Soucy [Starr 15302–B, 1927], Joseph Allard [Victor 263527–B, 1928] and Alfred Montmarquette [Starr 15475–A, 1928].)

A human-supplied musical analysis of these 59 strains identified 25 variants of the A strain, 15 variants of the B strain, 10 strains that were neither A nor B variants, and 9 strains that could be conceived of as distant variants of A (4 strains) or B (5 strains). In addition, this analysis revealed two types of B strains: those with an ascending melodic contour in the second bar (7 strains), and those with a descending figure (8 strains). We used our system to generate ranked lists for each of the 59 strains. For the 25 A strains, the top 24 results in the ranked list contained either 24 ($n=10$) or 23 ($n=15$) of the remaining A strains. The B-strain results were more complex and are summarized in figure 6.

These results suggest that the A-strain variants of "Money Musk" are more similar to each other than are the B-strain variants, and that B-ascending variants are more diverse than B-descending. This analysis also allows us to identify certain variants as musical outliers. As noted above, 15 of the A strains recalled 23 of 24 other A strains. In 12 of these instances, the missing strain was the same. This suggests that this strain would be a good candidate for further study.

The results in figure 6 also point to a split in the B-ascending strains, between those that are most similar to

the other B-ascending strains and those that are equally similar to B-ascending and B-descending strains. In particular, the strain recorded by Arthur-Joseph Boulay stands out for its dissimilarity to other B-ascending strains. These B-strain results suggests that the original musicological analysis that classified the B strains as either ascending or descending may need to be refined with reference to additional significant features.

B-ascending strains (n=7) for the tune "Money Musk"

Artist, year (strain)	Number of other B strains (n=14) in top 14 items on ranked list. This includes both B-ascending and B-descending strains.	Number of other B-ascending strains (n=6)...	
		..in top 14 items on list	..in top 6 items on list
W. Boivin, 1974 (B)	13	5	1
W. Boivin, 1974 (B')	13	5	2
W. Boivin, 1974 (B'')	6	6	4
A.-J. Boulay, 1923 (B)	8	0	0
U. Potvin, 1980 (B)	7	5	5
U. Potvin, 1980 (B')	12	5	5
I. Soucy, 1927 (B)	9	5	5

B-descending strains (n=8) for the tune "Money Musk"

Artist, year (strain)	Number of other B strains (n=14) in top 14 items on ranked list (includes B-ascending and B-descending).	No. of other B-descending strains (n=7)...	
		..in top 14 items on list	..in top 7 items on list
J. Allard, 1928 (B)	11	7	6
G. Joyal, 1956 (B)	11	7	7
G. Joyal, 1956 (B')	11	7	7
G. Lajoie, 1951 (B)	12	7	5
G. Lajoie, 1951 (B')	10	7	5
A. Montmarquette, 1928 (B)	10	7	6
É. Picard, 1930 (B)	10	7	5
A. Richard, 1975 (B)	11	7	5

Figure 6: Results for 15 B strains of "Money Musk" out of a database of 59 strains.

Examples such as these suggest that our approach may help scholars of instrumental dance music achieve a more nuanced study of musical similarity. Specifically, our system may help to identify concordances, parse degrees of melodic variation, and pinpoint instances that require further examination. The system also provides tools—the similarity matrices and the ranked lists—to facilitate such examination.

Certain instances of comparison remain problematic, however. The system does not currently recognize changes in meter, occasionally resulting in incipits that are slightly shorter or longer than four measures. This is the case for both "Reel princesse" and "Reel du bon vieux temps" (figure 1). The ranked list results for such cases are still reasonably accurate (figure 4). Note also that the placement of the barlines in metrically irregular renditions of tunes is at the discretion of the transcriber.

In addition, the French-Canadian repertoire contains some tunes with variants in both compound meter (9/8 or 6/8) and simple meter (3/2, 4/4, 2/2, or 2/4). In such cases, the system does not always find a satisfactory alignment between strong beats.

The system may also generate incorrect results when the two incipits are of different lengths. This is largely because variations in length of musically similar strains are due to an expansion of the shorter to the longer. While a naïve dynamic programming approach to alignment is insensitive to expansion, this issue may be solved by reducing the weighting on alignments that compress the shorter incipit.

7. FUTURE WORK

Although our system is currently designed for the specific attributes of French-Canadian fiddle tunes, the comparison functions and weighting calculation may be adapted for other repertoires. Our system may be particularly useful for repertoires in which new melodies are constructed using modified segments of extant melodies. Such repertoires are primarily aural, but may also include notated repertoires such as Renaissance Masses based on pre-existent material. This would require modifying the system for polyphonic sources. More immediately, we would like to investigate the application of our system to British Isles and North America fiddling traditions. We do not anticipate needing to revise the comparison functions and weighting calculation for this repertoire, and thousands of these tunes are already available in symbolic notation via online databases [3; 15; 18].

Our system identifies nuances between two strains and is particularly useful for identifying strains with a high degree of variation. However, we recognize that our approach may be less versatile than a more generalized comparison function such as an edit distance or Earth Mover's Distance function. Eventually we may seek to combine our system with a "first pass" edit distance or Earth Mover's Distance function.

8. CONCLUSION

Musical repertoires that circulate primarily in aural tradition often contain significant variance between different instances of the same tune. Analyzing variation and transformation in such repertoires has been an important part of ethnomusicology, musicology and folklore scholarship for decades. This paper has presented a novel tool to aid researchers in variance analysis in instrumental dance tunes.

The source code for our system has been published under an open source license, available on GitHub at <http://github.com/ELVIS-Project/fiddle-tunes>.

We believe that our system may be of practical use for musicologists and musicians specializing in the traditional instrumental musics of the British Isles and North America. It may also prove a useful model when building analytical tools for other repertoires containing a large number of variants.

ACKNOWLEDGEMENTS

The authors would like to thank the following people for their help and assistance: Ichiro Fujinaga, David Brackett, Bryan Duggan, Matt Kelly, Peter van Kranenburg, Dorothea Blostein, Marc Bolduc, and Jean Duval. This work was supported by the Social Sciences and Humanities Research Council of Canada as part of the Single Interface for Music Score Searching and Analysis (SIMSSA) Project, and by a Doctoral Award from Bibliothèque et Archives nationales du Québec.

9. REFERENCES

- [1] Antilla, C., and J. Cumming: "The VIS Framework: Analyzing counterpoint in large datasets." In *Proc. Conf. Int'l Society for Music Information Retrieval*. Taipei, 71–6. 2014.
- [2] S. Bayard: "Prolegomena to a study of the principal melodic families of British-American folk song." *The Journal of American Folklore* 63 (247) 1–44, 1950.
- [3] J. Chambers: "JC's ABC tune finder." 2015. <http://trillian.mit.edu/~jc/cgi/abc/tunefind> (accessed 3 May 2015).
- [4] J. Cowdery: "A fresh look at the concept of tune family." *Ethnomusicology* 28 (3) 495–504, 1984.
- [5] Cuthbert, M., and C. Ariza: "music21: A toolkit for computer-aided musicology and symbolic music data." In *Proc. Conf. of the Int'l Society for Music Information Retrieval*. Utrecht, The Netherlands, 637–42. 2010.
- [6] B. Duggan: "Machine annotation of traditional Irish dance music." PhD diss., Dublin Institute of Technology, 2009.
- [7] K. Dunlay: "'Correctness' in Cape Breton fiddle music." *MacKinnon's Brook: Traditional Fiddle Music of Cape Breton, Volume 4*. Rounder Records 7040 Liner Notes, 2008.
- [8] J. Duval: "Porteurs de pays à l'air libre: jeu et enjeux des pièces asymétriques dans la musique traditionnelle du Québec." PhD diss., Université de Montréal, 2013.
- [9] E. Favreau: "La transmission de la musique traditionnelle par la radio." *Bulletin Mnémo* 2 (1) 1997. [http://www.mnemo.qc.ca/html2/97\(11\)a.html](http://www.mnemo.qc.ca/html2/97(11)a.html) (accessed 3 May 2015).
- [10] Goertzen, C.: "Texas contest fiddling: What modern variation technique tells us." In *Routes & roots: Fiddle and dance studies from around the North Atlantic 4*, edited by I. Russell, and C. Goertzen, 98–111. Aberdeen: The Elphinstone Institute. 2012.
- [11] C. Gore: *The Scottish fiddle music index: The 18th & 19th century printed collections*. Musselburgh, Scotland: Amaisig. 1994.
- [12] — — —: "The Scottish music index." 2015. <http://www.scottishmusicindex.org> (accessed 3 May 2015).
- [13] L. Hart, and G. Sandell: *Danse ce soir: Fiddle and accordion music of Québec*. Pacific, MO: Mel Bay. 2001.
- [14] IREPI: "Yvon Mimeault, violoneux." *L'Inventaire des ressources ethnologiques du patrimoine immatériel* 2015. <http://www.irepi.ulaval.ca/fiche-yvon-mimeault-86.html>
- [15] J. Keith: "The session." 2015. <https://thesession.org> (accessed 3 May 2015).
- [16] M. Kelly: "Evaluation of melody similarity measures." MSc diss., Queen's University (Ontario), 2012.
- [17] Kleinberg, J., and É. Tardos: "Dynamic programming." In *Algorithm design*, 251–335. Boston: Pearson/Addison-Wesley. 2006.
- [18] A. Kuntz, and V. Pelliccioni: "Traditional tune archive." 2015. <http://tunearch.org/wiki/TTA> (accessed 3 May 2015).
- [19] G. Labbé: *Musiciens traditionnels du Québec (1920–1993)*. Montréal: VLB. 1995.
- [20] Library and Archives Canada: "The virtual gramophone." 2015. <http://www.collectionscanada.gc.ca/gramophone/index-e.html> (accessed 3 May 2015).
- [21] M. Mongeau, and D. Sankoff: "Comparison of musical sequences." *Computers and the Humanities* 24161–75, 1990.
- [22] M. Ó Súilleabháin: "Innovation and tradition in the music of Tommie Potts." PhD diss., Queen's University (Belfast), 1987.
- [23] L. Ornstein: "A life of music: History and repertoire of Louis Boudreault, traditional fiddler from Chicoutimi, Quebec." MA diss., Université Laval, 1985.
- [24] R. Spottswood: *Ethnic music on records: A discography of ethnic recordings produced in the United States, 1893–1942. Vol. 1: Western Europe*. Urbana: University of Illinois. 1991.
- [25] R. Thérien: *L'Histoire de l'enregistrement sonore au Québec et dans le monde 1878–1950*. Québec: Presses de l'Université Laval. 2003.
- [26] P. Van Kranenburg: "A computational approach to content-based retrieval of folk song melodies." PhD diss., Utrecht University, 2010.

A SEMANTIC-BASED APPROACH FOR ARTIST SIMILARITY

Sergio Oramas¹, Mohamed Sordo², Luis Espinosa-Anke³, Xavier Serra¹

¹Music Technology Group, Universitat Pompeu Fabra

²Center for Computational Science, University of Miami

³TALN Group, Universitat Pompeu Fabra

{sergio.oramas, luis.espinosa, xavier.serra}@upf.edu, msordo@miami.edu

ABSTRACT

This paper describes and evaluates a method for computing artist similarity from a set of artist biographies. The proposed method aims at leveraging semantic information present in these biographies, and can be divided in three main steps, namely: (1) entity linking, i.e. detecting mentions to named entities in the text and linking them to an external knowledge base; (2) deriving a knowledge representation from these mentions in the form of a semantic graph or a mapping to a vector-space model; and (3) computing semantic similarity between documents. We test this approach on a corpus of 188 artist biographies and a slightly larger dataset of 2,336 artists, both gathered from Last.fm. The former is mapped to the MIREX Audio and Music Similarity evaluation dataset, so that its similarity judgments can be used as ground truth. For the latter dataset we use the similarity between artists as provided by the Last.fm API. Our evaluation results show that an approach that computes similarity over a graph of entities and semantic categories clearly outperforms a baseline that exploits word co-occurrences and latent factors.

1. INTRODUCTION

Artist biographies are a big source of musical context information and have been previously used for computing artist similarity. However, only shallow approaches have been applied by computing word co-occurrences and thus the semantics implicit in text have been barely exploited. To do so, semantic technologies, and more specifically Entity Linking tools may play a key role to annotate unstructured texts. These are able to identify named entities in text and disambiguate them with their corresponding entry in a knowledge base (e.g. Wikipedia, DBpedia or BabelNet).

This paper describes a method for computing semantic similarity at document-level, and presents evaluation results in the task of artist similarity. The cornerstone of this work is the intuition that semantifying and formaliz-

ing relations between entity mentions in documents (both at in-document and cross-document levels) can represent the relatedness of two documents. Specifically, in the task of artist similarity, this derives in a measure to quantify the degree of relatedness between two artists by looking at their biographies.

Our experiments start with a preprocessing step which involve Entity Linking over artist biographical texts. Then, a knowledge representation is derived from the detected entities in the form of a semantic graph or a mapping to a vector-space model. Finally, different similarity measures are applied to a benchmarking dataset. The evaluation results indicate that some approaches presented in this paper clearly outperform a baseline based on shallow word co-occurrence metrics. Source code and datasets are available online¹.

The remainder of this article is structured as follows: Section 2 reviews prominent work in the fields and topic relevant to this paper; Section 3 details the different modules that integrate our approach; Section 4 describes the settings in which experiments were carried out together with the evaluation metrics used; Section 5 presents the evaluation results and discusses the performance of our method; and finally Section 6 summarizes the main topics covered in this article and suggests potential avenues for future work.

2. RELATED WORK

Music artist similarity has been studied from the score level, the acoustic level, and the cultural level [9]. This work is focused on the latter approach, and more specifically in text-based approaches. Literature on document similarity, and more specifically on the application of text-based approaches for artist similarity is discussed next.

The task of identifying similar text instances, either at sentence or document level, has applications in many areas of Artificial Intelligence and Natural Language Processing [17]. In general, document similarity can be computed according to the following approaches: surface-level representation like keywords or n-grams [6]; corpus representation using counts [28], e.g. word-level correlation, jaccard or cosine models; Latent factor models, such as Latent Semantic Analysis [8]; or methods exploiting external



© Sergio Oramas¹, Mohamed Sordo², Luis Espinosa-Anke³, Xavier Serra¹.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Sergio Oramas¹, Mohamed Sordo², Luis Espinosa-Anke³, Xavier Serra¹. "A Semantic-based Approach for Artist Similarity", 16th International Society for Music Information Retrieval Conference, 2015.

¹<http://mtg.upf.edu/downloads/datasets/semantic-similarity>

knowledge bases like ontologies or encyclopedias [12].

The use of text-based approaches for artist and music similarity was first applied in [7], by computing co-occurrences of artist names in web page texts and building term vector representations. By contrast, in [30] term weights are extracted from search engine’s result counts. In [33] n-grams, part-of-speech tagging and noun phrases are used to build a term profile for artists, weighted by employing tf-idf. Term profiles are then compared and the sum of common terms weights gives the similarity measure. More approaches using term weight vectors have been developed over different text sources, such as music reviews [11], blog posts [4], or microblogs [29]. In [18] Latent Semantic Analysis is used to measure artist similarity from song lyrics. Domain specific ontologies have also been applied to the problem of music recommendation and similarity, such as in [5]. In [16], paths on an ontological graph extracted from DBpedia are exploited for recommending music web pages. However, to the best of our knowledge, there are scant approaches in the music domain that exploit implicit semantics and enhance term profiles with external knowledge bases.

3. METHODOLOGY

The method proposed in this paper can be divided in three main steps, as depicted in Fig 1. The first step performs entity linking, that is the detection of mentions to named entities in the text and their linking to an external knowledge base. The second step derives a semantically motivated knowledge representation from the named entity mentions. This can be achieved by exploiting natural language text as anchor between entities, or by incorporating semantic information from an external knowledge base. In the latter case, a document is represented either as a semantic graph or as a set of vectors projected on a vector space, which allows the use of well known vector similarity metrics. Finally, the third step computes semantic similarity between documents (artist biographies in our case). This step can take into consideration semantic similarity among entity mentions in document pairs, or only the structure and content of the semantic graph.

The following sections provide a more detailed description of each one of these steps, along with all the approaches we have considered in each step.

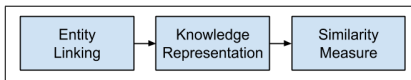


Figure 1. Workflow of the proposed method.

3.1 Entity Linking

Entity linking is the task to associate, for a given candidate textual fragment, the most suitable entry in a reference Knowledge Base (KB) [23]. It encompasses similar subtasks such as Named Entity Disambiguation [2], which

is precisely linking mentions to entities to a KB, or Wikification [21], specifically using Wikipedia as KB.

We considered several state-of-the-art entity linking tools, including Babelfy [23], TagMe [10], Agdistis [32] and DBpedia Spotlight [20]. However we opted to use the first one for consistency purposes, as in a later step we exploit *SensEmbed* [13], a vector space representation of concepts based on BabelNet [24]. Moreover, the use of a single tool across approaches guarantees that the evaluation will only reflect the appropriateness of each one of them, and in case of error propagation all the approaches will be affected the same.

Babelfy [23] is a state-of-the-art system for entity linking and word sense disambiguation based on non-strict identification of candidate meanings (i.e. not necessarily exact string matching), together with a graph based algorithm that traverses the BabelNet graph and selects the most appropriate semantic interpretation for each candidate.

3.2 Knowledge representation

3.2.1 Relation graph

Relation extraction has been defined as the process of identifying and annotating relevant semantic relations between entities in text [15]. In order to exploit the semantic relations between entities present in artist biographies, we applied the method defined in [25] for relation extraction in the music domain. The method basically consists of three steps. First, entities are identified in the text by applying entity linking. Second, relations between pairs of entities occurring in the same sentence are identified and filtered by analyzing the structure of the sentence, which is obtained by running a syntactic parser based on the formalism of dependency grammar [1]. Finally, the identified entities and relations are modeled as a knowledge graph. This kind of extracted knowledge graphs may be useful for music recommendation [31], as recommendations can be conveyed to users by means of natural language. We apply this methodology to the problem of artist similarity, by creating a graph that connects the entities detected in every artist biography. We call this approach RG (relation graph). Figure 2 shows the output of this process for a single sentence.

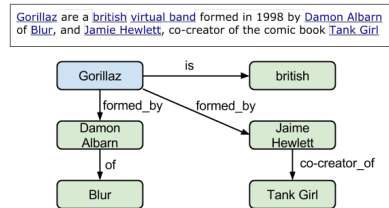


Figure 2. Relation graph of a single sentence

3.2.2 Semantically enriched graph

A second approach is proposed using the same set of linked entities. However, instead of exploiting natural language text, we use semantic information from the referenced knowledge base to enrich the semantics of the linked entities. We follow a semantic enrichment process similar to the one described in [27]. We use semantic information coming from DBpedia². DBpedia resources are generally classified using the DBpedia Ontology, which is a shallow, cross-domain ontology based on the most common infoboxes of Wikipedia. DBpedia resources are categorized using this ontology among others (e.g. Yago, schema.org) through the `rdfs:type` property. In addition, each Wikipedia page may be associated with a set of Wikipedia categories, which link articles under a common topic. DBpedia resources are related to Wikipedia categories through the property `dcterms:subject`.

We take advantage of these two properties to build our semantically enriched graph. We consider three types of nodes for this graph: 1) artist entities obtained by matching the artist names to their corresponding DBpedia entry; 2) named entities detected by the entity linking step; and 3) Wikipedia categories associated to all the previous entities. Edges are then added between artist entities and the named entities detected in their biographies, and between entities and their corresponding Wikipedia categories. For the construction of the graph, we can select all the detected named entities, or we can filter them out according to the information related to their `rdfs:type` property. A set of six types was selected, including *artist*, *band*, *work*, *album*, *musicgenre*, and *person*, which we consider more appropriate to semantically define a musical artist.

From the previous description, we define five variants of this approach. The first variant, which we call AEC (Artists-Entities-Categories), considers all 3 types of nodes along with their relations (as depicted in Figure 3). The second variant, named AE (Artists-Entities) ignores the categories of the entities. The third and fourth variant, named AEC-FT and AE-FT, are similar to the first and second variant, respectively, except that the named entities are filtered using the above mentioned list of 6 entity types. Finally, the fifth variant, EC, ignores the artist entities of node type 1.

3.2.3 Sense embeddings

The semantic representation used in this approach is based on SensEmbed [13]. SensEmbed is a vector space semantic representation of words similar to word2vec [22], where each vector represents a BabelNet synset and its lexicalization. Let A be the set of artist biographies in our dataset. Each artist biography $a \in A$ is converted to a set of disambiguated concepts Bfy_a after running Babelify over it.

²<http://dbpedia.org>

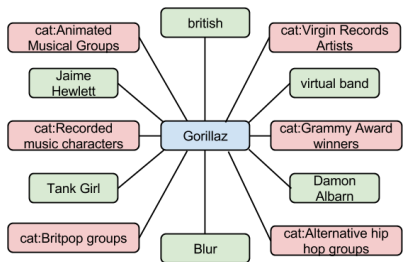


Figure 3. Semantically enriched subgraph of the same sentence from Figure 2, variant AEC with $h=1$

3.3 Similarity approaches

3.3.1 SimRank

SimRank is a similarity measure based on a simple graph-theoretic model [14]. The intuition is that two nodes are similar if they are referenced by similar nodes. In particular we use the definition of bipartite SimRank [14]. We build a bipartite graph with named entities and their corresponding Wikipedia categories (the EC variant from Section 3.2.2). The similarity between two named entities (say p and q) is computed with the following recursive equation:

$$s(p, q) = \frac{C}{|O(p)||O(q)|} \sum_{i=1}^{|O(p)|} \sum_{j=1}^{|O(q)|} s(O_i(p), O_j(q)) \quad (1)$$

where O denotes the out-neighboring nodes of a given node and C is a constant between 0 and 1. For $p = q$, $s(p, q)$ is automatically set up to 1. Once the similarity between all pairs of entities is obtained, we proceed to calculate the similarity between pairs of artists (say a and b) by aggregating the similarities between the named entities identified in their biographies, as shown in the following formula:

$$sim(a, b) = Q(a, b) \frac{1}{N} \sum_{e_a \in a} \sum_{e_b \in b} s(e_a, e_b) \quad \text{if } s(e_a, e_b) \geq 0.1 \quad (2)$$

where s denotes the SimRank of entities e_a and e_b and N is the number of (e_a, e_b) pairs with $s(e_a, e_b) \geq 0.1$. This is done to filter out less similar pairs. Finally, $Q(a, b)$ is a normalizing factor that accounts for the pairs of artists with more similar entity pairs than others.

3.3.2 Maximal common subgraph

Maximal common subgraph (MCS) is a common distance measure on graphs. It is based on the maximal common subgraph of two graphs. MCS is a symmetric distance metric, thus $d(A, B) = d(B, A)$. It takes structure as well as content into account. According to [3], the distance between two non empty graphs G_1 and G_2 is defined as

$$d(G_1, G_2) = 1 - \frac{|mcs(G_1, G_2)|}{\max(|G_1|, |G_2|)} \quad (3)$$

It can also be seen as a similarity measure s , assuming that $s = 1 - d$, as applied in [19]. To compute this similarity measure we need to have a graph for each artist. This can be achieved by finding subgraphs in the graph approaches defined in Section 3.2. A subgraph will include an artist entity node and its neighboring nodes. Furthermore, we apply the notion of h-hop item neighborhood graph defined in [26] to a semantic graph. Let $G = (E, P)$ be an undirected graph where E represent the nodes (entities), and P the set of edges with $P \subseteq E \times E$. For an artist item a in G , its h-hop neighborhood subgraph $G^h(a) = (E^h(a), P^h(a))$ is the subgraph of G formed by the set of entities that are reachable from a in at most h hops, according to the shortest path. Following this approach, we obtain an h-hop item neighborhood graph for each artist node of the semantic graph. Then, maximal common subgraph is computed between each pair of h-hop item neighborhood graphs. For each artist, the list of all similar artists ordered from the most similar to the less one is finally obtained.

3.3.3 Cumulative cosine similarity

For each pair of concepts $c \in \text{Bfy}_a$ and $c' \in \text{Bfy}'_a$ (as defined in Section 3.2.3), we are interested in obtaining the similarity of their closest senses. This is achieved by first deriving the set of associated vectors V_c and $V_{c'}$ for each pair of concepts c , c' , and then optimizing

$$\max_{v_c \in V_c, v_{c'} \in V_{c'}} \left(\frac{v_c \times v_{c'}}{\|v_c\| \|v_{c'}\|} \right) \quad (4)$$

i.e. computing cosine similarity between all possible senses (each sense represented as a vector) in an all-against-all fashion and keeping the highest scoring similarity score for each pair. Finally, the semantic similarity between two artist biographies is simply the average among all the cosine similarities between each concept pair.

4. EXPERIMENTAL SETUP

To evaluate the accuracy of the proposed approaches we designed an experimental evaluation over two datasets. The first dataset contains 2,336 artists and it is evaluated using the list of similar artists provided by the Last.fm API as a ground truth. The second dataset contains 188 artists, and it is evaluated against user similarity judgments from the MIREX Audio Music Similarity and Retrieval task. Apart from the defined approaches, a pure text-based approach for document similarity is added to act as a reference for the obtained results.

4.1 Datasets

4.1.1 Last.fm dataset

A dataset of 2,336 artist biographies was gathered from Last.fm. The artists in this dataset share a set of restrictions. Their biography has at least 500 characters and is

written in English. All of the artists have a correspondent Wikipedia page, and we have been able to mapped it automatically, obtaining the DBpedia URI of every artist. For every artist, we queried the getSimilar method of the Last.fm API and obtained an ordered list of similar artists. Every artist in the dataset fulfills the requirement of having at least 10 similar artists within the dataset. We used these lists of similar artists as the ground truth for our evaluation.

4.1.2 MIREX dataset

To build this dataset, the gathered artists from Last.fm were mapped to the MIREX Audio Music Similarity task dataset. The AMS dataset (7,000 songs from 602 unique artists) contains human judgments of song similarity. According to [29], the similarity between two artists can be roughly estimated as the average similarity between their songs. We used the same approach in [29], that is, two artists were considered similar if the average similarity score between their songs was at least 25 (on a fine scale between 0 and 100).

After the mapping, we obtained an overlap of 268 artists. As we want to evaluate Top-10 similarity, every artist in the ground truth dataset should have information of at least 10 similar artists. However, not every artist in the MIREX evaluation dataset fulfills this requirement. Therefore, after removing the artists with less than 10 similars, we obtained a final dataset of 188 artists, and used it for the evaluation.

4.2 Baseline

In order to assess the goodness of our approaches, we need to define a baseline approach with which to compare to. The baseline used in this paper is a classic vector-based model approach used in many Information Retrieval systems. A text document is represented as a vector of word frequencies (after removing English stopwords and words with less than 2 characters), and a matrix is formed by aggregating all the vectors. The word frequencies in the matrix are then re-weighted using TF-IDF, and finally latent semantic analysis (LSA) [8] is used to produce a vector of concepts for each document. The similarity between two documents can be obtained by using a cosine similarity over their corresponding vectors.

4.3 Evaluated approaches

From all possible combinations of knowledge representations, similarity measures and parameters, we selected a set of 10 different approach variants. The prefixes AEC, RG and AE refer to the graph representations (see Sections 3.2.1 and 3.2.2). SE refers to the sense embeddings approach, and LSA to the latent semantic analysis baseline approach. When these prefixes are followed by FT, it means that the entities in the graph have been filtered by type. The second term in the name refers to the similarity measure. MCS refers to maximal common subgraph, and SimRank and Cosine to SimRank and cumulative cosine similarity measures. MCS approaches are further followed by a number indicating the number of h-hops of the neighborhood subgraph.

Approach variants	Genres							Overall
	Blues	Country	Edance	Jazz	Metal	Rap	Rocknroll	
Ground Truth	5.78	5.46	6.88	7.04	7.10	8.68	5.17	6.53
LSA	4.43	4.12	3.80	4.64	5.79	5.08	4.74	4.69
RG MCS 1-hop	2.63	3.50	1.50	2.95	4.00	2.54	1.70	2.68
RG MCS 2-hop	4.14	4.92	1.69	2.80	3.78	3.06	2.77	3.27
AE MCS	5.52	5.15	4.36	7.00	4.34	5.36	4.46	5.11
AE-FT MCS	5.43	6.12	4.16	6.20	6.32	5.36	3.77	5.26
AEC MCS 1-hop	7.22	5.92	5.24	7.12	5.48	6.92	4.86	6.02
AEC MCS 2-hop	4.22	3.69	4.56	6.20	4.55	4.64	4.09	4.54
AEC-FT MCS 1-hop	6.91	6.80	6.04	7.60	6.79	7.12	5.37	6.59
AEC-FT MCS 2-hop	4.09	4.36	5.56	6.72	4.39	4.16	3.77	4.67
EC SimRank	6.74	5.38	3.16	6.40	4.59	4.44	3.80	4.85
SE Cosine	3.39	5.50	5.32	5.16	4.31	5.36	4.31	4.75

Table 3. Average genre distribution of the top-10 similar artists using the MIREX dataset. In other words, on average, how many of the top-10 similar artists are from the same genre as the query artist. LSA stands for Latent Semantic Analysis, RG for Relation Graph, SE for Sense Embeddings, and AE, AEC and EC represent the semantically enriched graphs with Artists-Entities, Artist-Entities-Categories, and Entities-Categories nodes, respectively. As for the similarity approaches, MCS stands for Maximum Common Subgraph.

Approach variants	Precision@N		nDCG@N	
	N=5	N=10	N=5	N=10
LSA	0.100	0.169	0.496	0.526
RG MCS 1-hop	0.059	0.087	0.465	0.476
RG MCS 2-hop	0.056	0.101	0.433	0.468
AE MCS	0.106	0.178	0.503	0.517
AE-FT MCS	0.123	0.183	0.552	0.562
AEC MCS 1-hop	0.120	0.209	0.573	0.562
AEC MCS 2-hop	0.086	0.160	0.550	0.539
AEC-FT MCS 1-hop	0.140	0.218	0.588	0.578
AEC-FT MCS 2-hop	0.100	0.160	0.527	0.534
EC SimRank	0.097	0.171	0.509	0.534
SE Cosine	0.095	0.163	0.454	0.484

Table 1. Precision and normalized discounted cumulative gain for Top-N artist similarity using the MIREX dataset ($N=\{5, 10\}$)

4.4 Evaluation measures

To measure the accuracy of the artist similarity we adopt two standard performance metrics such as Precision@N, and nDCG@N (normalized discounted cumulative gain). Precision@N is computed as the number of relevant items (i.e., true positives) among the top-N items divided by N, when compared to a ground truth. Precision considers only the relevance of items, whilst nDCG takes into account both relevance and rank position. Denoting with s_{ak} the relevance of the item in position k in the Top-N list for the artist a , then nDCG@N for a can be defined as:

$$\text{nDCG@N} = \frac{1}{\text{IDCG@N}} \sum_{k=1}^N \frac{2^{s_{ak}} - 1}{\log_2(1 + k)} \quad (5)$$

where IDCG@N indicates the score obtained by an ideal or perfect Top-N ranking and acts as a normalization factor. We run our experiments for $N = 5$ and $N = 10$.

Approach variants	Precision@N		nDCG@N	
	N=5	N=10	N=5	N=10
LSA	0.090	0.088	0.233	0.269
RG MCS 1-hop	0.055	0.083	0.126	0.149
AE MCS	0.124	0.200	0.184	0.216
AE-FT MCS	0.136	0.201	0.224	0.260
AEC MCS 1-hop	0.152	0.224	0.277	0.297
AEC-FT MCS 1-hop	0.160	0.242	0.288	0.317

Table 2. Precision and normalized discounted cumulative gain for Top-N artist similarity using the Last.fm dataset ($N=\{5, 10\}$)

5. RESULTS AND DISCUSSION

We evaluated all the approach variants described in Section 4.3 on the MIREX dataset, but only a subset of them on the Last.fm dataset, due to the high computational cost of some of the approaches.

Table 1 shows the Precision@N and nDCG@N results of the evaluated approaches using the MIREX dataset, while Table 2 shows the same results for the Last.fm dataset. We obtained very similar results in both datasets. The approach that gets best performance for every metric, dataset and value of N is the combination of the Artists-Entities-Categories graph filtered by types, with the maximal common subgraph similarity measure using a value of $h = 1$ for obtaining the h-hop item neighborhood graphs.

Furthermore, given that the MIREX AMS dataset also provides genre data, we analyzed the distribution of genres in the top-10 similar artists for each artist, and averaged them by genres. The idea is that an artist's most similar artists should be from the same genre as the seed artist. Table 3 presents the results. Again, the best results are obtained with the approach that combines the Artists-Entities-Categories graph filtered by types, with the maxi-

mal common subgraph similarity measure using a value of $h = 1$ for the h-hop item neighborhood graphs.

We extract some insights from these results. First, semantic approaches are able to improve pure text-based approaches. Second, using knowledge from an external knowledge base provides better results than exploiting the relations inside the text. Third, using a similarity measure that exploits the structure and content of a graph, such as maximal common subgraph, overcomes other similarity measures based on semantic similarity among entity mentions in document pairs.

6. CONCLUSION

In this paper we presented a methodology that exploits semantic technologies for computing artist similarity, which can be divided in three main steps: First, named entity mentions are identified in the text and linked to a knowledge base. Then, these entity mentions are used to construct a semantically motivated knowledge representation. Finally a similarity function is defined on top of the knowledge representation to compute the similarity between artists. For each one of these steps we explored several approaches, and evaluated them against a small dataset of 188 artist biographies, and a larger dataset of 2,336 artists, both obtained from Last.fm.

Results showed that a combination of the Artists-Entity-Categories graph filtered by types, and a maximal common subgraph similarity measure using a value of $h = 1$ for obtaining the h-hop item neighborhood graphs, clearly outperforms a baseline approach that exploits word co-occurrences and latent factors. In the light of these results, the following conclusions can be drawn: First, semantic approaches may outperform pure text-based approaches. Second, we observe that knowledge leveraged from external ontologies may improve the accuracy of the similarity measure. Third, reducing noise by filtering linked entities by type is a rewarding step that contributes to an improved performance. Finally, we show that similarity measures that take into consideration the structure and content of a graph representation may achieve much higher performance.

There are still many avenues for future work. We would like to compare our semantic-based approach with acoustic and collaborative filtering approaches. In addition, the use of text sources different from artist biographies can be studied. Finally, in order to improve the results obtained by our semantic approach, different state-of-the-art entity linking tools can be applied, or a specific entity linking tool for the music domain could be created for this purpose.

7. REFERENCES

- [1] Bernd Bohnet. Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 89–97, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.
- [2] Razvan Bunescu and Marius Pasca. Using Encyclopedic Knowledge for Named Entity Disambiguation. In *Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics (EACL-06)*, pages 9–16, Trento, Italy, 2006.
- [3] Horst Bunke and Kim Shearer. A graph distance metric based on the maximal common subgraph. *Pattern Recognition Letters*, 19(3-4):255–259, March 1998.
- [4] Óscar Celma, Pedro Cano, and Perfecto Herrera. Search Sounds An audio crawler focused on weblogs. In *7th International Conference on Music Information Retrieval (ISMIR)*, 2006.
- [5] Óscar Celma and Xavier Serra. FOAFing the music: Bridging the semantic gap in music recommendation. *Web Semantics*, 6:250–256, 2008.
- [6] Hung Chim and Xiaotie Deng. Efficient phrase-based document similarity for clustering. *Knowledge and Data Engineering, IEEE Transactions on*, 20(9):1217–1229, 2008.
- [7] William W. Cohen and Wei Fan. Web-collaborative filtering: recommending music by crawling the Web. *Computer Networks*, 33:685–698, 2000.
- [8] Scott C. Deerwester, Susan T Dumais, Thomas K. Landauer, George W. Furnas, and Richard A. Harshman. Indexing by latent semantic analysis. *JASIS*, 41(6):391–407, 1990.
- [9] Daniel P. W. Ellis, Brian Whitman, Adam Berenzweig, and Steve Lawrence. The quest for ground truth in musical artist similarity. In *Proc. International Symposium on Music Information Retrieval (ISMIR 2002)*, pages 170–177, 2002.
- [10] Paolo Ferragina and Ugo Scaiella. Tagme: on-the-fly annotation of short text fragments (by wikipedia entities). In *Proceedings of the 19th ACM international conference on Information and knowledge management*, pages 1625–1628. ACM, 2010.
- [11] X Hu, JS Downie, Kris West, and AF Ehmman. Mining Music Reviews: Promising Preliminary Results. In *ISMIR*, pages 536–539, 2005.
- [12] Xiaohua Hu, Xiaodan Zhang, Caimei Lu, Eun K Park, and Xiaohua Zhou. Exploiting Wikipedia as external knowledge for document clustering. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 389–396. ACM, 2009.
- [13] Ignazio Iacobacci, Mohammad Taher Pilehvar, and Roberto Navigli. Sensembled: Enhancing word embeddings for semantic similarity and relatedness. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Beijing, China, July 2015. Association for Computational Linguistics.

- [14] Glen Jeh and Jennifer Widom. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 538–543. ACM, 2002.
- [15] J. Jiang and C. Zhai. A systematic exploration of the feature space for relation extraction. In *HLT-NAACL*, pages 113–120, 2007.
- [16] José Paulo Leal, Vânia Rodrigues, and Ricardo Queirós. Computing Semantic Relatedness using DBpedia. *1st Symposium on Languages, Applications and Technologies, SLATE 2012*, 2012.
- [17] Hongzhe Liu and Pengfei Wang. Assessing Text Semantic Similarity Using Ontology. *Journal of Software*, 9(2):490–497, 2014.
- [18] Beth Logan and Daniel P W Ellis. Toward Evaluation Techniques for Music Similarity. *SIGIR 2003: Workshop on the Evaluation of Music Information Retrieval Systems*, pages 7–11, 2003.
- [19] Mathias Lux and Michael Granitzer. A Fast and Simple Path Index Based Retrieval Approach for Graph Based Semantic Descriptions. In *Proceedings of the Second International Workshop on Text-Based Information Retrieval*, 2005.
- [20] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia spotlight: shedding light on the web of documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8. ACM, 2011.
- [21] Rada Mihalcea and Andras Csomai. Wikify!: linking documents to encyclopedic knowledge. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 233–242. ACM, 2007.
- [22] Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic Regularities in Continuous Space Word Representations. In *HLT-NAACL*, pages 746–751, 2013.
- [23] Andrea Moro, Francesco Ceconi, and Roberto Navigli. Multilingual word sense disambiguation and entity linking for everybody. In *Proceedings of the 13th International Conference on Semantic Web (P&D)*, 2014.
- [24] Roberto Navigli and Simone Paolo Ponzetto. BabelNet: Building a very large multilingual semantic network. In *Proceedings of the 48th annual meeting of the association for computational linguistics*, pages 216–225. Association for Computational Linguistics, 2010.
- [25] Sergio Oramas, Mohamed Sordo, and Luis Espinosa-anke. A Rule-Based Approach to Extracting Relations from Music Tidbits. In *2nd Workshop in Knowledge Extraction from Text, WWW'15*, 2015.
- [26] Vito Claudio Ostuni, Tommaso Di Noia, Roberto Mirizzi, and Eugenio Di Sciascio. A Linked Data Recommender System using a Neighborhood-based Graph Kernel. *15th International Conference on Electronic Commerce and Web Technologies*, pages 1–12, 2014.
- [27] Vito Claudio Ostuni, Sergio Oramas, Tommaso Di Noia, Xavier Serra, and Eugenio Di Sciascio. A Semantic Hybrid Approach for Sound Recommendation. *24th International World Wide Web Conference (WWW 2015)*, pages 3–4, 2015.
- [28] Mark Rorvig. Images of similarity: A visual exploration of optimal similarity metrics and scaling properties of TREC topic-document sets. *Journal of the American Society for Information Science*, 50(8):639–651, 1999.
- [29] Markus Schedl, David Hauger, and Julián Urbano. Harvesting microblogs for contextual music similarity estimation: a co-occurrence-based framework. *Multimedia Systems*, 20(6):693–705, 2013.
- [30] Markus Schedl, Peter Knees, and Gerhard Widmer. A Web-Based Approach to Assessing Artist Similarity using Co-Occurrences. In *Proceedings of the 4th International Workshop on Content-Based Multimedia Indexing (CBMI'05)*, 2005.
- [31] Mohamed Sordo, Sergio Oramas, and Luis Espinosa. Extracting Relations from Unstructured Text Sources for Music Recommendation. In *20th International Conference on Applications of Natural Language to Information Systems*, pages 1–14, 2015.
- [32] Ricardo Usbeck, Axel-Cyrille Ngonga Ngomo, Sören Auer, Daniel Gerber, and Andreas Both. Agdistis-agnostic disambiguation of named entities using linked open data. In *International Semantic Web Conference*, page 2, 2014.
- [33] B. Whitman and S. Lawrence. Inferring descriptions and similarity for music from community metadata. In *Proceedings of the 2002 International Computer Music Conference*, pages 591–598, 2002.

PREDICTING PAIRWISE PITCH CONTOUR RELATIONS BASED ON LINGUISTIC TONE INFORMATION IN BEIJING OPERA SINGING

Shuo Zhang, Rafael Caro Repetto, Xavier Serra
Music Technology Group, Universitat Pompeu Fabra
ssz6@georgetown.edu, {rafael.caro, xavier.serra}@upf.edu

ABSTRACT

The similarity between linguistic tones and melodic pitch contours in Beijing Opera can be captured either by the contour shape of single syllable units, or by the pairwise pitch height relations in adjacent syllable units. In this paper, we investigate the latter problem with a novel machine learning approach, using techniques from time series data mining. Approximately 1300 pairwise contour segments are extracted from a selection of 20 arias. We then formulate the problem as a supervised machine-learning task of predicting types of pairwise melodic relations based on linguistic tone information. The results give a comparative view of fixed and mixed-effects models that achieved around 70% of maximum accuracy. We discuss the superiority of the current method to that of the unsupervised learning in single-syllable-unit contour analysis of similarity in Beijing Opera.

1. INTRODUCTION

One of the most salient aspects of Chinese operas is the role of various dialects and their distinct tone contours. In the musicological study of Beijing opera, the similarity between linguistic tone contours of the lyrics and the vocal melodic contours is a classic problem that arises from the nature of Chinese tone languages. In a tone language, as opposed to an intonation language, the pitch contour of a speech sound (often a syllable) can be used to distinguish lexical meaning. In singing, however, such pitch contour can be overridden by the melody of the music, making the lyrics difficult to decode by listeners [1]. In order for lyrics to be more intelligible, Beijing opera's melody is traditionally arranged with considerations of lyrics tone information. The degree and manner of this incorporation, however, is only partly known through scholarly work [1-4]. The difficulty of this problem is further complicated by the fact that there are two dialects with distinct tone contours within Beijing opera (Beijing and HuGuang dialects, or BJ and HG in this paper) [3].

Previous works cited above indicate that the similarity between linguistic tones and melodic pitch contours in Beijing Opera can be captured either by the contour shape of single syllable units, or by the pairwise pitch height

relations in adjacent syllable units. [1] considered the single-syllable unit contour analysis with a time-series data mining approach. This study concluded that while the Smoothing Spline model's R-squared values are consistent with the expected variance relations between the first tone and other tones, overall there exists a large amount of un-explained variance in melodic contours that cannot be attributed to grouping of tone categories from a single tone system (BJ or HG).

In this paper we investigate the second type of similarity of linguistic tones and melodic contours. Following musical literature [12], we postulate that the perceived similarity of the melody to a tone category is realized by the similar pitch height relations in a pair of adjacent syllable units in singing (to that of the tone in speech). We then formulate this problem as a supervised machine learning problem of predicting the type of pairwise pitch height relations based on features derived from linguistic attributes. First we perform experimentation on the most efficient and cognitively accurate time-series representations for pitch contour vectors and extract the class labels. Following feature extraction and data preprocessing, a series of multinomial, binary and mixed effects regression models are trained. These allow us to progressively achieve our two main goals: First, using linguistic information to predict (with improved accuracy) the melodic pairwise pitch height relations; second, as a consequence, we also obtain a better understanding of the effect of various linguistic and other attributes on the types of pitch height relations observed in Beijing opera.

The remainder of the paper is organized as follows. Section 2 gives the formulation of the pairwise tone-melody similarity as a supervised machine learning problem, followed by the description of data collection and preprocessing in Section 3. The core methodologies of time-series data representation experimentation and model training are described in Section 4. Section 5 and 6 discuss the results, including the comparison of models and interpretation of model parameters.

2. PROBLEM FORMULATION

Recent research revealed that tone identification by humans does not necessarily depend on the availability of full tone contour information [5]. In the light of this finding, pairwise tone-melody similarity is therefore a cognitively plausible way for the melody to convey underlying tone information without being fully similar to the contour of the linguistic tone. For example, a high-level tone



(5-5)¹ followed by a low-rise tone (2-4) can be reflected in melody as long as the perceived starting pitch of the second syllable is lower than the first. Perceptually, the beginning position of a syllable is the most salient, being a prominent position that carries much phonetic information such as formant transitions [6]. Alternatively, one may propose that this relation can be reflected by the ending pitch of the first syllable and the beginning pitch of the second syllable, being the closest pair in time. Less plausible is the case where this similarity is reflected in the ending region pitch of both syllables.

We therefore formulate this problem of pairwise similarity as a supervised machine learning problem. First, we define three types of relations between the two syllables in proximity (mostly adjacent, but can be separated by a short instrumental interlude), based on the relative pitch height: ascending (A), descending (D), and level (L). These are our target class labels. Second, we define three subtypes of pairwise similarity based on the location of the similarity: Onset-Onset (BB), Offset-Onset (EB), and Offset-Offset (EE). We will train a separate model for each type of similarity. Third, we formulate the research objective: given linguistic tone and other attributes of a pair of syllables in the lyrics, can we correctly predict the type of relations of relative pitch height in vocal melody (A, D, or L as class label)?

3. DATA COLLECTION

3.1 Data Collection

The current study uses about 1300 syllable-sized contours extracted from a selection of 20 arias in a pre-segmented and annotated Beijing opera audio collection corpus [7]. Each syllable in this data set is annotated with linguistic tone, word, artist, role type, melodic type (*shengqiang*), rhythmic type (*banshi*), and relevant metadata information. This set is selected according to a number of criteria: (1) we selected only *yuanban*, a rhythmic type in which the duration of a syllable sized unit bears the most similarity to that of speech; (2) we selected both types of *shengqiang*, namely *xipi* and *erhuang*; (3) we selected five role types: D(dan), J(jing), LD(laodan), LS(laosheng), and XS(xiaosheng). For each combination of *shengqiang* and role types, we selected two arias, yielding a total of 20 arias for analysis. This set of arias is selected by a music scholar with expertise in Beijing opera music (who is the second author), and is therefore a representative set that is both comprehensive and selective for the task of studying the tone-melody relationship.

3.2 Pitch Contour Extraction

The fundamental frequency of vocal melodic contours is computed using the MELODIA [10] package

within the Essentia audio signal-processing library in Python [11], in order to minimize the interference of background instrumental ensemble to the computation of F0 of the primary vocal signal. All rows of F0 values associated with a specific pitch contour are automatically assigned a unique pitch contour id that encodes the aria, tone, and temporal order information of the syllable. For the sake of analysis, we produce down-sampled 30-point F0 vectors by using equidistant sampling across each pitch contour. A 5-point weighted averaging sliding window is applied to smooth the signal. The single-syllable contour data is then converted into a pairwise-syllable contour data file where each row has 60 pitch points of the two adjacent syllable contours, plus other attributes.

4. METHODOLOGY

4.1 Time-series representation

First we perform automatic extraction of our target class labels (A, D, or L). In order to capture the accurate perceived pitch heights in the beginning and ending regions of each syllable-sized melodic contour, we first convert the 30-point pitch contour into a lower dimension representation using the Symbolic Aggregate approximation (SAX) [8]. SAX transforms the pitch contour into a symbolic representation using Piecewise Aggregate Approximation with a user-designated length (*nseg*, or sometimes referred to as word size, is the desired length of the feature vector) and alphabet size (*alpha*), the latter being used to divide the pitch space of the contour into *alpha* parts assuming a Gaussian distribution of F0 values.

Using this technique, we experiment with the parameter settings of SAX with the goal of yielding the most similar relation types as a human listener would judge it. To perform this experiment, we first have a human listener annotate a selection of 260 sample tone contours extracted from our audio collection². For each contour, the listener would rate the type of pairwise relation (A, D or L) by listening through the contour pairs. The experiment is proctored automatically by a Praat Script program and the presentation of each pair is separated by a white noise of 5 seconds. The listener rates all 260 contours consecutively.

Next, we permute the single syllable contour unit parameter values³ within intervals *nseg* ([3,8]) and *alpha* ([3,6]). Each combination of the parameters yields a SAX representation for all contours and then three pairwise relation types (BB, EB, EE) based on the representation is extracted. We then compute the similarity / accuracy of each representation to the human judgments. Here, it is

² Human rater is used only to train the parameters on a smaller sample so that we can perform automatic label extraction on larger scales of data.

³ To ensure the consistency of pitch space division with the Gaussian breakpoints, we convert a pair of syllables at a time, making the *nseg* parameter twice as big.

¹ The numbering here follows the relative pitch height from low to high: 1<2<3<4<5.

noteworthy that the perceived beginning pitch height of a syllable-sized melodic contour does not necessarily correspond to a predetermined meaningful musical unit such as a note. This is due to the nature of the Beijing opera that has many fine melisma across the melody. In this case, we do not attempt to define a perceptually or musically grounded unit of ‘beginning pitch’, but rather we will let the experiment results decide which parameter configuration would be the closest to the human judgment. Since SAX is already a dimensionality reduction algorithm, we thereby define the beginning pitch of a syllable as the first symbol in the symbolic time-series representation. After the parameters are chosen, we use the SAX representations to extract pairwise relation types for the entire training set.

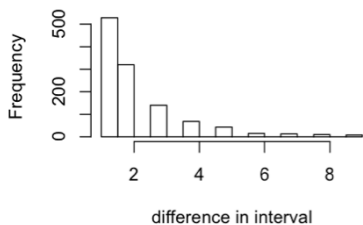


Figure 1 Histogram of interval distances between pairwise syllables

4.2 Regression modeling

4.2.1 Feature Extraction and Data Preprocessing

The basic set of features includes the attributes extracted from the corpus annotations: ToneFirst, ToneSecond, (of the first and second syllable in a pairwise contour), word, artist, role_type, shengqiang, banshi, as well as the duration of both syllables. All except the last one are nominal attributes.

We additionally extracted three sets of compound features based on linguistic tones: first, a toneCombination feature that encodes the particular tone combinations (e.g., tone1_tone2) of this pairwise contour; second, six other features that encodes the types of linguistic tone pitch height relations of this tone combination (BB, EB, and EE) as well as the two dialectal tone systems (BJ and HG). These features are therefore (BB_BJ, BB_HG, EB_BJ, EB_HG, EE_BJ, EE_HG). These are the only features that directly encode the types of pairwise linguistic tone relations into the feature vectors, using numbered pitch height system from linguistic literature (e.g., tone 3 in BJ is 214 and in HG is 42, 1<2<3<4). Theoretically these features should not be used all at once, since each one of our regression models would only account for one type of output relations (BB, EB, or EE). However, since the previous studies suggest that the BJ and HG tone systems are likely intermixed in affecting the output melodic contour[1], we include both of these two types of features

in each model. A third feature encodes the temporal distance/ number of interval segments between the pair of syllable: it is hypothesized that a closer pair of syllable would contribute to the manifestation of linguistic information. We have eliminated those pairs whose distance is greater than 10 intervals (the intervals in between a pair could be due to various reasons, but mostly likely instrumental interlude). Figure 1 shows the distribution of distance in units of time intervals in the entire data set (where an interval is a syllable unit in our data segmentation). From this distribution we can see that most pairs are sung consecutively.

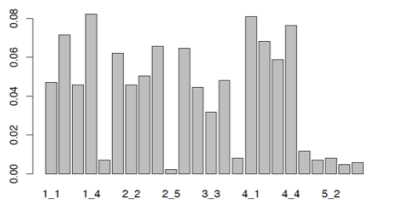


Figure 2 Barplot of the frequency of tone combination types in training data, where x_y indicates tone x to tone y combination

We perform several measures of data preprocessing on the training data. First, we eliminate all contours whose syllable duration is longer than a threshold of 5s (based on the distribution of durations). This is based on the observation that if the syllable is too long, the temporal relations are sparser and it has more chance to musically embellish its contours, further obscuring the linguistic information. Second, we observe that there is an extreme imbalance between tone categories 1-4 and tone 5. Linguistically, tone 5 is a ‘neutral’ tone that carries different contours according to their context. Figure 2 shows this imbalance in the toneCombination feature. We eliminated all examples with tone 5 in order to avoid singularity problems with generalized linear modeling.

Lastly, the output class label distribution is also highly skewed (Figure 3). This is an interesting property of this musical data set especially when compared with its expected counterpart in language (i.e., the set of six tone features that encodes pairwise tone pitch relations). Figure 3 plots the set of pairwise musical pitch relation labels (BB, EB, EE) alongside its expected counterpart (BB_BJ, EB_BJ, etc.) linguistic tone feature distributions. It is noteworthy that not coincidentally, the linguistic pairwise types have a quite uniform distribution whereas the musical pairwise types have a very skewed distribution, with the ‘L(evel)’ label being the rare class. This is probably a product of music: music, being a play largely about the manipulation of pitch, is intentionally avoiding many of the adjacent syllables (or notes) starting or ending with the same pitch height. Therefore in these cases we may hypothesize that the music is overriding linguistic configurations, thus obscuring our model. For this reason as well as motivations from the machine learning perspective, we created a second data set where all ‘L’ labels are removed from the training data (which

is a small portion, ref. Figure 3). Therefore we use this second data set for binary logistic and mixed-effects regression modeling in the latter part of the study.

4.2.2 Multinomial Regression

Our first model approximates this problem with a multinomial logistic regression using the original data set with three output class labels (A, D, L). The multinomial logistic regression is an extension to the binary logistic regression modelling, where we train one-versus-other models for each of the class labels. The model outputs the probability of assigning each label and selects the label with the highest probability as the predicted label.

For all of the algorithms used in this study, as previously discussed, we build three models assuming three different types of relations (BB, EB, EE). We first build baseline multinomial logistic regression models with all available basic features. Then we incrementally drop features whose coefficients are insignificant and having low predicting powers and end up with the best model for this setting. This set of features is used throughout the rest of the models in conjunction with compound features.

4.2.3 Binary (Fixed Effects) Logistic Regression

We then perform all subsequent regression modeling on the binary data set. As a baseline for this data set, we perform classic fixed effects binary logistic regression and compare the result with a number of well known machine learning algorithms such as Support Vector Machine (SVM), decision tree (J48 in Weka), Neural Networks, and Naïve Bayes.

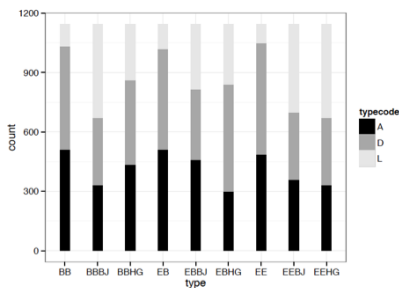


Figure 3 Distribution of pairwise pitch relation types across class labels (BB, EB, EE) and corresponding expected linguistic feature distributions

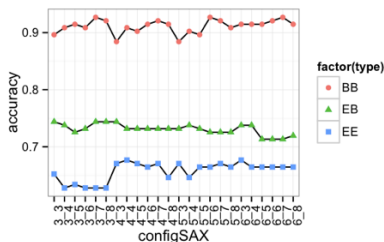


Figure 4 Accuracy of pairwise pitch relation type prediction by different SAX parameters (`alpha_nseg`) with z-transform

4.2.4 Mixed Effects Logistic Regression

A mixed-effects regression model performs prediction by combining the contributions from fixed effects and random effects. Parameters associated (coefficients) with the particular levels of a covariate are known as the “effects” of the levels. Essentially, if the set of possible levels of the covariate is fixed and reproducible we model the covariate using fixed-effects parameters. If the levels that we observed represent a random sample from the set of all possible levels we incorporate random effects in the model [9].

We extend from generalized linear models (GLMs) to multilevel GLMs by adding a stochastic component \mathbf{Z} to the linear predictor (see (1)), where the random effects vector \mathbf{b} is normally distributed with mean 0 and variance-covariance matrix Σ . In a mixed-effects logistic regression model, we plug the stochastic linear predictor into the binomial (logistic) linking function.

$$\eta = \alpha + \beta_1 X_1 + \dots + \beta_n X_n + b_0 + b_1 Z_1 + \dots + b_m Z_m + \epsilon \quad (1)$$

In the current setting, the random effects of our feature correspond to the variable words. Any general model would usually exclude the particular words of the two syllables as a fixed-effect feature; however, the problem with the fixed effects model (like the one described above) is that it is not capturing the variances in the output label caused by being different words. The mixed effects model corrects this by estimating the conditional mode of the random effect term coefficients \mathbf{B} . Strictly speaking, we don’t estimate the random effects in the same sense that we estimate model parameters. Instead, we consider the conditional distribution of B given the observed data, $(B|Y = y)$, where Y is the output class label [9].

All modeling in this study are done with 10-fold cross validation in the training phase.

5. RESULT

5.1 Time-series Representation Experiment Results

Overall the time-series representation experiments using SAX technique yielded informative results. First, as shown in Figure 4, the differences in the accuracy are mostly the effect of varying nseg, whereas the effect of alpha parameter is not apparent (except for the EE type). Second, somewhat surprisingly, the average type prediction accuracy vary significantly across the three types: BB>>EB>EE, with Onset-to-Onset relation types performing almost perfectly at peak accuracy of 93%. This contrast is surprising considering that the human listener (who is a skilled musician) did not rate these three types directly; instead, the listener only had to rate the beginning and end of a single syllable on a numbered fixed scale¹ and the values for the three types of relations were extracted using that reference scale automatically. This systematic difference in the accuracy of these three types of relations indicates that the Offset annotation / judgment has a much lower correlation with the acoustic signal than the Onset annotation. One possible explanation for this could be that the ending of a syllable is embellished with more melismas than the beginning portion of the syllable, making the correlation lower. However, that would not predict the systematic lower performance of SAX EB and EE, which is at a lower resolution than the original acoustic signal. An alternative explanation is that the offset position is not as salient as the onset position, making it less appropriate for the location of carrying linguistic tone information.

Due to the differences in performance, we choose the SAX parameter settings for each of the three types: {BB:6_7, EB: 4_3, BB: 6_3}, where the parameter combinations stand for alpha_nseg.

5.2 Regression Modeling Results²

	Coeff1(D)	p-value	Coeff2 (L)	p-value
Intercept(A)	-0.0761	6e-01	-1.2447	*2e-05
Duration1	0.2149	0.0437	0.2594	0.1048
T_1_2	-0.8454	*6e-06	-0.8159	*7e-03
T_1_3	-0.6952	*0.0004	-0.4844	0.1148
T_1_4	-1.2471	*6e-12	-0.9295	*1e-03
T_2_2	0.5765	*0.0018	-0.1137	0.6986
T_2_3	0.6709	*0.0007	0.2909	0.3265
T_2_4	1.1505	*2e-10	0.2175	4e-01

Table 1. Significant (basic) predictors overview from multinomial regression for BB type, with asterisks indicating coefficients significant at 0.05 level. Coef1 and coef2 represent the regression coefficient associated with features. T_{i_j} is the coefficient associated with a i-th tone being a tone j.

¹ Here, all the contours are extracted sequentially from the same aria so the judgment and extraction of consecutive relations are accurate.

² In this table, class A is treated as base/default level, and the Intercepts represent the base probabilities of D and L with regard to A without any knowledge about features.

First, results (Table 1) of multinomial logistic regression reveal that tone information and the duration of the first syllable are among the most significant predictors of the probability of a pairwise contour being one of the three output classes (A, D or L). Concretely, for example, being a Tone 2 in the first syllable would significantly lower the probability of being a “D” by log-odds -0.8454 or odds $\exp(-0.8434)$, and being a tone 4 for the second syllable would significantly increase the probability of being a “L” by log-odds 0.2175 or odds of $\exp(0.2175)$, where the $\exp()$ is the exponentiation function. Overall the multinomial regression has a mean classification accuracy of 56.7% for all types of models on the 10-fold cross validation on the entire data set. This is a lower baseline for the subsequent models. Due to the skewed output distribution of class labels, the model consistently assigns the lowest probability to “L” in all predictions.

Despite this finding, further analysis shows that the basic tone features (ToneFirst and ToneSecond) have limited predictive power compared to other compound tone features. Therefore in the subsequent analysis we drop these two basic features and keep the other 1+6 types of compound features.

Our binary logistic model on the binary class data set (class label A and D) improves the accuracy by about 9%. This result is comparable across different classification algorithms (Table 2).

Algorithm	mean Accuracy
Binary Logistic Regression	65.12%
Decision Tree (J48)	61.57%
SVM	62.44%
NaiveBayes	61.56%
NeuralNetwork	60.07%

Table 2. Average performance of different algorithms on the binary classification data set with a 10-fold cross validation

The mixed-effect model further improves the prediction accuracy to around and above 70%. This set of models has two variations. The first set is built with ToneCombination and the six other compound features (two per model) as well as the duration of the first and second syllable as fixed effects features, and the Word as a simple scalar random effect feature (1|Word). The second set includes more complex random effects features (1+duration|Word), which takes into account the interaction between the duration of the first syllable (fixed effect) and the Word (random effect) feature. The performance of these two sets varies between the three types of models. Table 3 gives a comprehensive overview of the evaluation of the models.

Overall, all models have shown that the prediction accuracy decreases from BB to EB to EE. This is in accordance with our initial SAX representation accuracy rank, therefore is expected. However, the underlying reason for this is unclear, as discussed previously.

Figure 5 gives an overview of the model performances based on average accuracy.

	AIC	BIC	LogLik	Se.Residual	Accuracy
BB1	1059.3	1153.1	-509.7	-0.438	70.77%
BB2	1063.1	1166.3	-509.5	-0.437	69.99%
EB1	1086.8	1180.6	-523.4	0.526	65.30%
EB2	1096.4	1193.6	-523.2	0.512	65.8%
EE1	1104.3	1198.1	-532.2	0.685	60.95%
EE2	1107.2	1210.4	-531.6	0.649	64.80%

Table 3 Model comparison for mixed effect models, where AIC and BIC are commonly used Akaike Information Criterion and Bayesian Information Criterion

6. DISCUSSION AND CONCLUSION

The current study has considered the problem of the similarity between linguistic tones and melodic contours in Beijing opera in the form of pairwise pitch height relations. We formulate this similarity problem as a supervised machine learning problem of predicting the type of relations based on linguistic tone information. We have shown that using a set of linguistic features alone (tone and duration information), the model is able to achieve the best average accuracy of around 65% to 71% based on the types of hypothesized relations, after we have reduced the output class labels to binary (for reasons discussed above). Here we discuss several aspects of the interpretation and evaluation of the current results.

First, the performance of the models has shown consistently that Onset-Onset is a more robust pairwise relation type compared to Offset-Onset and Offset-Offset. However, this result may be dependent upon the initial performance rank of SAX representation accuracy for these three types. To better understand this phenomenon, we performed a post-hoc re-analysis of the SAX conversion using the original fundamental frequency data without down-sampling and evaluate its accuracy. The result showed a more balanced yet overall lower accuracy on extracted class labels as compared to human annotation (75%, 72%, and 78% peak accuracy values for BB, EB, and EE). When using this set of class labels, we obtained generally lower performance on the best mixed effects models in the classification task (57%, 68%, and 58% for BB, EB, and EE). Noticeably, the BB type model has a lower prediction accuracy than the EB type, making the Offset-to-onset relation more robust. Meanwhile, there is less confidence in this result due to the general lower accuracy in the representation of the class labels.

Second, we should also bear in mind that in the current problem, the class labels of pitch height relations are dependent upon the musical considerations on top of the linguistic considerations. For all practical and theoretical reasons we believe that Beijing opera music has its own rules that at many times take precedence over linguistic rules, and that should give us a large proportion of unexplained variances when predicting pairwise pitch relations. Considering this factor, it is fair to conclude that the current models have shown effectively the high degree of pairwise similarity between linguistic tones and

melodic contours in Beijing opera. For the same reason discussed above, we have justified our decision to take the “L” class out from our model because of its likely irrelevance to linguistic information (and should be explained by musical considerations).

Third, comparing the current study with previous works on the single-syllable contour similarity [1], we observe that the current approach yields higher explanatory power than the previous approach, while requiring significantly less computing resources.¹ Specifically, it is worth noting that while the contour-shape-based SSANOVA models in [1] suffers from the lack of knowledge on the exact weights of the two dialects (BJ and HG), the current approach is able to encode expected pairwise pitch relations from both dialects into the features, thus making it more effective in a supervised learning task.

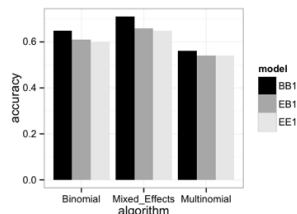


Figure 5: Model accuracy overview: binomial, mixed-effects, multinomial models

7. ACKNOWLEDGEMENT

This research was funded by the European Research Council under the European Union Seventh Framework Program, as part of the CompMusic Project (ERC grant agreement 267583).

8. REFERENCES

- [1] S. Zhang, R. Caro Repetto, S. Xavier: “Study of the similarity between linguistic tones and melodic pitch contours in Beijing Opera singing”. *Proceedings of The 15th International Society for Music Information Retrieval (ISMIR) Conference*, pp.345-348. Taiwan, October 27-31 2014.
- [2] Pian, R.C.: *Text Setting with the Shipyi Animated Aria*. In *Words and Music: The Scholars View*, edited by Laurence Berman, 237270. Cambridge: Harvard University Press, 1972.
- [3] Xu, Z.: 2007. *Jiu shi nian lai jingju de sheng diao yan jiu zhi hui gu*. (Review of Ninety Years of Research

¹ This comment is in reference to the expensive computations of the DTW distance matrices in the time-series data mining of the pitch contours in a large dataset. In addition, the SSANOVA model in [1] was only able to achieve a R-squared value of around 0.2, suggesting low explanatory power of the variance present in the data.

- of tones in Beijing Opera). *Nankai Journal of Linguistics*, 10(2):39-50.
- [4] Stock, J: A Reassessment of the Relationship Between Text, Speech Tone, Melody, and Aria Structure in Beijing Opera. *Journal of Musicological Research* (18:3): 183206. 1999. [16]
- [5] Lai, Yuwen and Jie Zhang. : Mandarin lexical tone recognition: the gating paradigm. In Emily Tummons and Stephanie Lux (eds.), *Proceedings of the 2007 Mid-America Linguistics Conference, Kansas Working Papers in Linguistics* 30. 183-194.2008.
- [6] Steriade, Donca. (2001). Directional asymmetries in place assimilation. In *The role of speech perception in phonology*, eds. Elizabeth Hume and Keith Johnson, 219–250. San Diego: Academic Press.
- [7] C. Repetto and X. Serra X. “Creating a Corpus of Jingju (Beijing Opera) Music and Possibilities for Melodic Analysis”. *Proceedings of the 15th International Society for Music Information Retrieval Conference*, Oct. 27th-31st 2014, Taipei (Taiwan).
- [8] Lin,J., Keogh,E.,Wei,L.,and Lonardi,S.: Experiencing SAX: a novel symbolic representation of time series. *Data Mining and Knowledge Discovery*. Oct.2007, Vol.15, Issue.2, pp107-144.2007.
- [9] D.Bates. *Mixed-Effects Modeling with R*. Springer: 2010.
- [10] Salamon, J and Gmez E: "Melody Extraction from Polyphonic Music Signals using Pitch Contour Characteristics", *IEEE Transactions on Audio, Speech and Language Processing*, 20(6):1759-1770.2012.
- [11] Bogdanov, D., Wack N., Gmez E., Gulati S., Herrera P., Mayor O., et al.: ESSENTIA: an Audio Analysis Library for Music Information Retrieval. *International Society for Music Information Retrieval Conference (ISMIR'13)*. 493-498.(2013).
- [12] Lian, B. *Xiqu zuoqu jiaocheng* (Textbook on Chinese opera composition). Shanghai Conservatory Publications, 1999.

SONG2QUARTET: A SYSTEM FOR GENERATING STRING QUARTET COVER SONGS FROM POLYPHONIC AUDIO OF POPULAR MUSIC

Graham Percival, Satoru Fukayama, Masataka Goto

National Institute of Advanced Industrial Science and Technology (AIST), Japan

graham@percival-music.ca, s.fukayama@aist.go.jp, m.goto@aist.go.jp

ABSTRACT

We present Song2Quartet, a system for generating string quartet versions of popular songs by combining probabilistic models estimated from a corpus of symbolic classical music with the target audio file of any song. Song2Quartet allows users to add novelty to listening experience of their favorite songs and gain familiarity with string quartets. Previous work in automatic arrangement of music only used symbolic scores to achieve a particular musical style; our challenge is to also consider audio features of the target popular song. In addition to typical audio music content analysis such as beat and chord estimation, we also use time-frequency spectral analysis in order to better reflect partial phrases of the song in its cover version. Song2Quartet produces a probabilistic network of possible musical notes at every sixteenth note for each accompanying instrument of the quartet by combining beats, chords, and spectrogram from the target song with Markov chains estimated from our corpora of quartet music. As a result, the musical score of the cover version can be generated by finding the optimal paths through these networks. We show that the generated results follow the conventions of classical string quartet music while retaining some partial phrases and chord voicings from the target audio.

1. INTRODUCTION

Cover songs are arrangements of an original song with certain variations which add novelty. Changing the instruments used is one such variation, but a complete switch of instrumentation may result in very unusual parts. For example, completely replacing a chord-heavy guitar part with a violin may result in unplayable (or very difficult) chords. Arranging music for different instruments requires consideration about the music those instruments normally perform.

Previous approaches in automated arrangement are mostly performed in the symbolic domain of music. Melody harmonization and re-harmonization of chord sequences take symbols of chords or pitches as inputs [1, 7, 10, 16]. Guitar arrangements of piano music can be generated from a

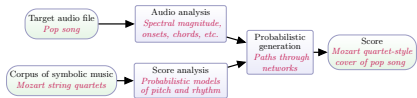


Figure 1: Generating a cover song with a specific style. Sample results are available at:

<https://staff.aist.go.jp/m.goto/Song2Quartet/>

MusicXML score [14]. Statistical modelling of a corpus has also been used to generate electronic dance music [6]. Furthermore, automatically generating music in a specific instrumental style is not well explored. In a great deal of work on computer-assisted composition [8], some automatic composition systems attempted to generate results with a particular composer’s musical style [4] or the user’s musical style [15]. However, those systems cannot be used to generate cover songs in a particular instrument style by preserving the recognizable parts of the original songs.

We present Song2Quartet to address this issue. An overview of our system is shown in Figure 1. Two novel aspects of this work, the audio analysis for generating cover songs and generating music in a specific instrumental style, are addressed in the audio analysis and score analysis modules, respectively.

To ensure that the generated cover songs include features that are also recognizable in the original audio, the audio analysis module estimates notable rhythms, chord voicings, and contrary motions between melody and bass by extracting the audio spectrum. In parallel, to generate music to be playable and recognizably following the classical string quartet style, the score analysis module captures characteristics of the string quartet from the corpus of symbolic music such as the typical note onsets in a measure and the pitch transitions of each instrument in the quartet.

These two aspects are balanced by means of a probabilistic formulation, where the corpus style and audio analysis are combined by weighted multiplication. The audio analysis provides probabilities for observing note events at every 16th note, and the score analysis mainly provides the transition probabilities of notes. We formalize our generation of cover songs as finding the sequence of notes which maximizes the probabilities obtained from the modules using dynamic programming, with techniques to compress the search space to make our problem tractable.



© Graham Percival, Satoru Fukayama, Masataka Goto. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Graham Percival, Satoru Fukayama, Masataka Goto. “Song2Quartet: A System for Generating String Quartet Cover Songs from Polyphonic Audio of Popular Music”, 16th International Society for Music Information Retrieval Conference, 2015.

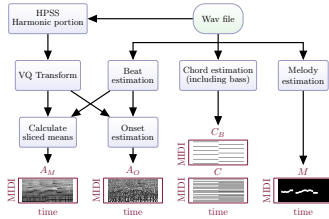


Figure 2: Audio analysis (4 measures shown in examples).

2. ANALYSIS

2.1 Features needed from audio

Knowing which pitches are in the polyphonic music is useful in creating cover songs. Since multi-pitch analysis methods often suffer from pitch and onset detection errors when handling polyphonic music with a drum track, we cannot simply apply the analysis beforehand and use the analysis results as constraints. However, we can use the audio feature extraction portions of multi-pitch analysis to aid in generating cover songs. Concretely, after performing Harmonic/Percussive source separation, the magnitudes and onsets of each note are obtained by applying a variable-Q spectral transform and calculating the salience function of the onset events.

The melody, chords, bass, and beats of a song provide musical facets which should be observed in the cover version of a song. These facets are extracted from the audio using Songle, a music understanding engine [13]. The melody and bass pitches, as well as the chord labels, are segmented according to the time grid provided by the analyzed beats. Later, these will be combined with the beat-aligned audio spectral analysis to form probabilistic constraints.

2.2 Audio analysis

Figure 2 shows an overview of the audio analysis. We perform Harmonic/Percussive source separation with median filtering [9], then use a variable-Q transform (VQT) [18] with a Blackman-Harris window and the variable-Q parameter γ set to use a constant fraction of the equivalent rectangular bandwidths [11], giving us spectral analysis S . The frequency range was set to 65 Hz–2500 Hz (MIDI pitches 36–99).

We then perform beat estimation on the original audio with Songle and divide each beat into 4, giving 16^{th} notes. The means (over time) of VQT bins that fall within the range of each 16^{th} note are calculated, producing the sliced spectrogram A_M . A_M is normalized to the range $[0, 1]$.

To estimate onset probabilities in the target song, we use two methods: flux of A_M and first-derivative Savitzky-Golay filtering [17] on S . The flux of A_M is simply the half-wave rectified difference between successive 16^{th} notes of A_M . For the latter method, we calculate the smoothed first derivative of S along the time axis using Savitzky-Golay filtering with a window size of 21 samples to find

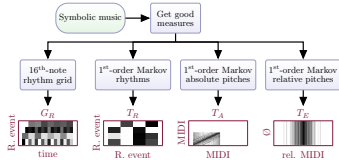


Figure 3: Score analysis (Mozart cello in examples).



Figure 4: Rhythmic events detected in the score.

the peaks of S . To quantize the onset estimation to the 16^{th} -note level, we find the maximum peak within a time window equal to a 16^{th} note duration, but shifted backwards in time by 25% to accommodate slight inaccuracies in the beat detection. Both methods operate on each MIDI pitch independently. We set A_O to be the sum of the two methods, and normalize it to the range $[0, 1]$.

Finally, we extract two more pieces of information using [13]: the melody M , and the chords in each song, including both the overall chord name C and the bass pitch C_B .

2.3 Features needed from the score

Features obtained from the score analysis contribute to maintaining the musical style. Classical string quartet music rarely includes complex rhythms and very large pitch intervals, so we obtain these tendencies as probabilities of rhythm and pitch intervals from the corpus of scores.

2.4 Score analysis

Figure 3 shows an overview of the score analysis. We used the Music21 [5] toolkit and corpus to analyze string quartets by Haydn, Mozart, and Beethoven. Our analysis comprised of pitches and rhythms, and only used music in 4/4 time which fit into a 16^{th} -note grid. If the time signature changed in the middle of a movement, we only considered the portion(s) in 4/4.

We calculated the probabilities of rhythmic events in a 16^{th} note grid. Rhythmic events were defined as one of four possible values: 0 indicated a new note, 1 indicated a new rest, 2 indicated a continued note, and 3 indicated a continued rest; an example is shown in Figure 4. This resulted in a 4×16 matrix of probabilities G_R , with each probability being the number of occurrences divided by the number of measures.

We extracted 1st-order Markovian [2] rhythm transitions. This is simply the probability of each [previous event, next event] pair occurring, and produced a 4×4 matrix T_R .

We calculated 1st-order Markovian pitch transitions for both absolute pitches and relative pitches. We considered a chord-note or pair of chords to include every pitch transition between the notes in successive chords. For simplicity, we recorded these transitions in two 100×100 matrices T_A

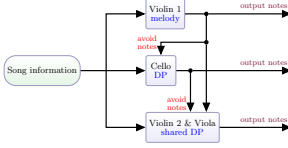


Figure 5: Overview of creating parts.

and T_E , even though a classical string quartet will not have any notes below MIDI pitch 36. For the absolute pitches, we added a 10^{-3} chance of any transition between valid pitches; this is necessary to allow some modern pop songs with non-classical chord progressions to be generated, particularly in the cello which is limited to the bass notes C_B .

3. PROBABILISTIC GENERATION

Figure 5 gives an overview of generating the quartet parts. First, the violin 1 part is set to the melody. Second, the cello part is generated with a probabilistic method and dynamic programming. Third, the violin 2 and viola parts are generated together via the same probabilistic method and dynamic programming.

To prepare for the dynamic programming, we need to define the *emission* and *transition* matrices, denoted by E and T , respectively. Our time unit is 16^{th} notes, and we consider 200 possible events for each time-slice: 0 is a rest, 1–99 are note onsets of the same MIDI pitches, 100 is a held rest, and 101–199 are held notes (of MIDI pitch +100). We define N as the number of 16^{th} notes in the target song. An overview of calculating E and T is shown in Figure 6.

3.1 Constructing probability matrices

3.1.1 Construction emission probabilities E

The emission probabilities E is a matrix of size $N \times 200$, representing every possible event at every 16^{th} note. They are generated by calculating E_O (onsets) and E_H (held notes), each of size $N \times 100$,

$$E_O = A'_O \otimes C' \otimes G'_R \otimes I_R \otimes V \quad (1)$$

$$E_H = A'_M \otimes C' \otimes G'_R \otimes I_R \otimes V \quad (2)$$

where \otimes is the element-wise product. The intuition behind this multiplication is that we consider each variable to be an independent probability distribution, so we are calculating the joint distribution. E_O and E_H are then stacked vertically to form E . The variables are:

- A'_O, A'_M — *Audio onsets and magnitudes*: Audio onsets A_O and magnitudes A_M for MIDI pitches 1–99 are taken directly from the audio analysis. The “silence” event (0) is set to a constant value of 10^{-5} .
- C' — *Chord tones*: We construct a matrix of all MIDI pitches for every 16^{th} note in the song; each cell is 1 if that pitch is in the given chord, 10^{-2} otherwise. For the cello, we use the bass note of each chord C_B ; for other instruments, we use any chord tone included in C .

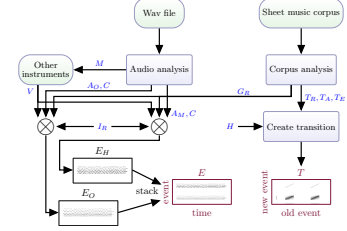


Figure 6: Calculating emission and transition probabilities E and T . \otimes indicates element-wise multiplication.

- G'_R — *Rhythm grids*: We take the overall probability of a rhythmic event in the corpus at each 16^{th} note G_R , and repeat it for every 16 time-slices in N .
- I_R — *Extreme instrument ranges*: We specify maximum bounds for instrument ranges: MIDI pitches 36–69 for cello, 48–81 for viola, and 55–99 for violin. When a corpus of symbolic music is used, the pitch transition probabilities narrow these ranges; I_R is only relevant if the user chooses not to use any corpus.
- V — *Avoid previously-used notes*: We reduce the probability of using the same notes as other instruments by setting them to 10^{-2} in V ; other events are set to 1. We also reduce the probability of playing a note one octave higher than an existing note (as those are likely octave errors in the audio analysis) by likewise setting those values to 10^{-2} .

We eliminate any non-rest values less than 10^{-3} to reduce the computational load for music generation.

3.1.2 Construction transition probabilities T

The transition probabilities T are a matrix of size 200×200 , representing every possible event-to-event pair.

$$T = T'_R \otimes T'_A \otimes T'_E \otimes H \quad (3)$$

The variables are:

- T'_R — *Rhythm transitions*: We use T_R , the probability of each rhythm event following a previous rhythm event. The note onset and held note probabilities are copied to vectors 1–99 and 101–199 respectively, while the rest onset and held rest probabilities are copied into vectors 0 and 100.
- T'_A, T'_E — *Pitch transitions*: We use the probabilities of each pitch following a previous pitch considering absolute or relative pitches, T_A and T_E respectively. These matrices are originally 100×100 ; we simply copy the matrices four times to create 200×200 matrices (that is to say, allowing these relative transitions to apply to onset-onset, onset-held, held-onset, held-held pairs).
- H — *Hold-events only after onset-events*: Each “hold” event (events 100 and up) can only occur after its respective “onset” event. We formalize this constraint as a matrix H where rows 0–99 are all 1, while rows 100–199 contain two identity matrices (in columns 0–99 and 100–199).

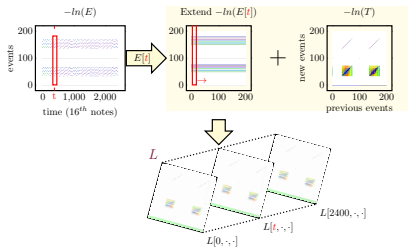


Figure 7: Combining emission and transition probabilities E and T into overall log-probabilities L .

3.2 Generation of a cover song under constraints

We combine E and T to form the log-probability L of the arranged score given the observed audio and corpus data, which has dimensions $N \times 200 \times 200$. For each $t \in N$,

$$L[t, :, :] = -\ln(E[t, :, :]) - \ln(T) \quad (4)$$

where $E[t, :, :]$ indicates that the 1×200 column vector $E[t]$ is extended to form a 200×200 matrix. This is illustrated in Figure 7. Since E and T contain very small numbers, we add their negative log-values instead of multiplying them.

L can be visualized by considering it to be a network of time-events (Figure 8). The maximum probability of a given score occurs when the negative log-probability is minimized; i.e. by finding the shortest path through L with a standard dynamic programming algorithm [3].

3.2.1 Local and Global Shortest Paths

As shown in Figure 5, we calculate the cello accompaniment part first. After that, we could solve the viola and then violin 2 parts separately, but we found that this occasionally produced very high violin 2 music. Instead we solve the violin 2 and viola parts together, with the constraint that they cannot play the same pitch at the same time.

In order to find two shortest paths simultaneously, we construct a large network with every possible combination of nodes from each time-slice of the individual violin 2 and viola networks. For example, if at time t the violin 2 could have 4 possible events and the viola could have 5 possible events, then the combined network will have 20 possible events for time t . The edge weights are simply the sum of the existing edges from the individual networks.

3.2.2 Compacting Matrices

To lower memory usage and improve processing time, we reduce the size of the matrices. We construct a mapping for each time-slice t between the non-infinite weights in L and a smaller matrix. This takes approximately 1 second, and results in a matrix which is roughly 1% of the original size (e.g., 96 million entries reduced to 1.2 million entries). Note that this compression is lossless and does not affect the shortest-path calculation, as an edge with weight ∞ will not appear in the shortest path.

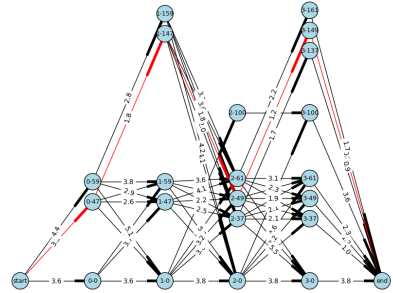


Figure 8: Network of possible pitches L ; shortest path colored red. Node labels are in the form “time-event”, with event x being a MIDI pitch onset ($x < 100$) or hold ($x \geq 100$). For legibility, edges with a weight of infinity and nodes with no non-infinite-weight edges are not displayed.

“Compacting” L in this way speeds up the computation of the single cello part, but its true value is found when combining the violin 2 and viola parts. Without any compacting, a normal pop song (150 measures) produces a network for a single part with $2400 \times 200 \times 200 = 9.6 \cdot 10^7$ entries. However, naively combining the violin 2 and viola parts produces a network with $2400 \times 200^2 \times 200^2 = 3.8 \cdot 10^{12}$ entries (15 TB of memory). We therefore perform two rounds of compacting; before and after combining the parts. After compacting the individual violin 2 and viola parts, we are left with networks of size approximately 1.6 million and 2.3 million. After performing the second round of compacting (this time on the combined matrix), the memory requirement is reduced from 5.8 GB to 0.25 GB.

3.2.3 Weighted probabilities

We found that the initial system produced music which was too heavily biased towards one “prototypical” measure of rhythms for each composer. We therefore multiplied each matrix by its own weighting factor, and allowed the user to specify and experiment with their own desired weights.

4. EXAMPLES AND DISCUSSION

To illustrate aspects of the generated music, we created a few cover versions of “PROLOGUE” (RWC-MDB-P-2001 No.7) from the RWC Music Database [12], with a variety of weights to the probability distributions. Short excerpts of the beginning of “PROLOGUE” are shown in Figure 10 with four variations: no corpus analysis, no audio spectral analysis, equal weights, and a set of custom weights.

Figures 10a and 10b clearly demonstrate the usefulness of combining audio with score analysis. Figure 10a does not use any corpus information (the weights of G_R , T_R , T_A , and T_E are set to 0), and produces music which is not idiomatic and is extremely difficult to perform. In the other extreme, Figure 10b uses the full Haydn string quartet corpus analysis, but does not use any spectral information

(the weights of A_O and A_M are set to 0), and produces music which is playable but very repetitive and “boring”: Other than measure 10 (the transition from the introduction to the melody), each instrument in the accompaniment plays the same rhythm in every measure (with the exception of the cello in measure 5), and 76% of measures contain a single pitch while 24% of measures contains two pitches.

Figure 10c uses all available data with weights of 1, and the music is both quite playable and more interesting than Figure 10b. There is more variation in the rhythms, and most notes are typical classical-style durations such as whole notes, half notes, or quarter notes. There are a few non-chord pitch changes (e.g., violin 2 measure 3, viola measure 13), but not many. This version contains one mistake: the viola in measure 13 begins with a $C\sharp$ 16th note which quickly changes to a C chord-tone. This could be avoided by decreasing the probability of non-chord tones, but doing so would also decrease the chance of a non-chord tone in the original song from being reproduced. This is an illustration of the choices available to the user.

Figures 10d (Haydn), 10e (Mozart), and 10f (Beethoven) demonstrate a custom set of weights. After some experimentation, we (subjectively) chose to set the onset A_O weight to 0.9, the corpus rhythms G_R and T_R weights to 0.5, and the corpus pitch transition T_A and T_E weights to 0.25. These three cover versions produce noticeably distinct music, arising solely due to the corpus used. The overall distribution of rhythmic durations seems natural: the cello has longer notes than the inner two voices. The distribution of pitches is reasonable, with all instruments playing in a comfortable range; the corpus clearly helps in avoiding the extreme pitches that were present in Figure 10a.

A few parts of the cover versions are the same in all compositions. Measure 10 always ends with a $G\sharp-C\sharp$ (alternatively “spelled” as $B\sharp$) in the cello and violin 2, with the viola filling in a transition from $D\sharp$ to $C\sharp$ (or $B\sharp$); this makes a nice V–I chord sequence ($G\sharp$ major to $C\sharp$ major) leading into measure 11. In addition, the V–I resolution in measures 10–11 always includes contrary motion in the cello and violin 2. Our probabilistic generation does not take relative motion of multiple voices into account, so this nice voice leading must arise from the strength of its presence in the audio spectral analysis.

A few problems exist in the voice leading. For example, Figure 10d shows a number of parallel fifths (e.g., viola and cello, measures $4 \rightarrow 5 \rightarrow 6$, $8 \rightarrow 9$). These likely arise due to the 2nd and 3rd harmonics of bass guitar notes in the original recording. A similar problem occurs with sudden jumps of an octave after one 16th or 8th note appears in a few places (e.g., viola measure 4 and cello measure 12). These also likely arise due to inaccuracies in the spectral analysis: the energy in upper partials of a single note can vary, so multiple onsets are detected in close succession. More advanced signal processing in terms of onset estimation or pitch salience calculation could mitigate this issue. Another fix for the parallel fifths would be to use a more advanced mathematical model; a first-order Markov model does not track the inter-dependence between quartet parts.

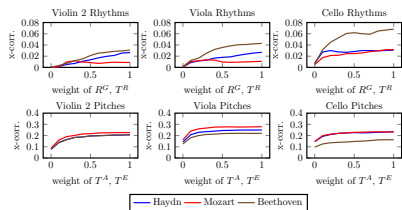


Figure 9: Objective analysis of weights; unless otherwise specified, our custom weights are used.

4.1 Objective analysis

Figure 9 shows the effect of changing the rhythmic or pitch corpus weights. The “pitch” plots show the cross-correlation between the corpus relative pitch distribution T_E and the relative pitches calculated from the generated scores. The “rhythm” plots show the cross-correlation between corpus and generated scores, based on the types of measures appearing in the output. Concretely, we construct a dictionary of full-measure rhythmic events (such as 0222133002130011 from Figure 4) along with their frequency of appearance, for both the corpus and the generated music. We then calculate the cross-correlation between those dictionaries for the corpus and each cover version.

Increasing the weight generally increases the correlation between corpus and generated music for both pitches and rhythms. One counter-example is violin 2 and viola in Mozart quartets. We theorize that this arises because increasing the rhythmic weight reduces the number of “completely eighth note” measures in the generated music, however such measures are very common in the original corpus.

5. CONCLUSION AND FUTURE WORK

We presented Song2Quartet, a system for generating string quartet cover versions of popular music using audio and symbolic corpus analysis. Both the target pop song audio file and the corpus of classical music contribute to the output; using only one or the other produces clearly inferior results. In order to avoid awkward second violin parts, we performed a semi-global optimization whereby we created the second violin and viola parts at the same time.

The current system makes a number of ad hoc assumptions, such as the melody always being played by the first violin and all rhythms fitting into 16th-note rhythms. Our evaluation was primarily based on informal listening, which showed promise despite some voice leading errors.

We plan to extend the data-driven corpus analysis so that users may generate cover versions for other groups of classical instruments. We also plan to add a GUI so that users can place the melody in different instruments at any point in the song. Finally, we would like to include evaluations of the generated scores’ “playability” by musicians.

Acknowledgments: This work was supported in part by CREST, JST.



Figure 10: Sample output; full versions and synthesized audio available at:

<https://staff.aist.go.jp/m.goto/Song2Quartet/>

6. REFERENCES

- [1] Moray Allan and Christopher K. I. Williams. Harmonising chorales by probabilistic inference. In *NIPS*, pages 25–32, 2005.
- [2] Charles Ames. The markov process as a compositional model: a survey and tutorial. *Leonardo*, pages 175–187, 1989.
- [3] Richard Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, USA, 1 edition, 1957.
- [4] David Cope. Computer modeling of musical intelligence in EMI. *Computer Music Journal*, pages 69–83, 1992.
- [5] Michael Scott Cuthbert, Christopher Ariza, and Lisa Friedland. Feature extraction and machine learning on symbolic music using the music21 toolkit. In *ISMIR*, pages 387–392, 2011.
- [6] Arne Eigenfeldt and Philippe Pasquier. Considering vertical and horizontal context in corpus-based generative electronic dance music. In *Proceedings of the Fourth International Conference on Computational Creativity*, volume 72, 2013.
- [7] Benjamin Evans, Satoru Fukayama, Masataka Goto, Nagisa Munekata, and Tetsuo Ono. AutoChorusCreator: Four-Part Chorus Generator with Musical Feature Control, Using Search Spaces Constructed from Rules of Music Theory. In *International Computer Music Conference*, 2014.
- [8] Jose D Fernández and Francisco Vico. Ai methods in algorithmic composition: A comprehensive survey. *Journal of Artificial Intelligence Research*, pages 513–582, 2013.
- [9] Derry Fitzgerald. Harmonic/Percussive Separation using Median Filtering. In *International Conference on Digital Audio Effects*, 2010.
- [10] Satoru Fukayama and Masataka Goto. Chord-sequence-factory: A chord arrangement system modifying factorized chord sequence probabilities. In *ISMIR*, pages 457–462, 2013.
- [11] Brian R Glasberg and Brian CJ Moore. Derivation of auditory filter shapes from notched-noise data. *Hearing research*, 47(1):103–138, 1990.
- [12] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. RWC Music Database: Popular, Classical and Jazz Music Databases. In *ISMIR*, volume 2, pages 287–288, 2002.
- [13] Masataka Goto, Kazuyoshi Yoshii, Hiromasa Fujihara, Matthias Mauch, and Tomoyasu Nakano. Songle: A web service for active music listening improved by user contributions. In *ISMIR*, pages 311–316, 2011.
- [14] Gen Hori, Hirokazu Kameoka, and Shigeki Sagayama. Input-Output HMM Applied to Automatic Arrangement for Guitars. *Journal of Information Processing*, 21(2):264–271, 2013.
- [15] Francois Pachet. The continuator: Musical interaction with style. *Journal of New Music Research*, 32(3):333–341, 2003.
- [16] Francois Pachet and Pierre Roy. Musical harmonization with constraints: A survey. *Constraints*, 01/2001(6):7–19, 2001.
- [17] Abraham. Savitzky and M. J. E. Golay. Smoothing and differentiation of data by simplified least squares procedures. *Analytical Chemistry*, 36(8):1627–1639, 1964.
- [18] Christian Schörkhuber, Anssi Klapuri, Nicki Holighaus, and Monika Dörfler. A matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution. In *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society, 2014.

EXPLORING DATA AUGMENTATION FOR IMPROVED SINGING VOICE DETECTION WITH NEURAL NETWORKS

Jan Schlüter and Thomas Grill

Austrian Research Institute for Artificial Intelligence, Vienna
jan.schlueter@ofai.at thomas.grill@ofai.at

ABSTRACT

In computer vision, state-of-the-art object recognition systems rely on label-preserving image transformations such as scaling and rotation to augment the training datasets. The additional training examples help the system to learn invariances that are difficult to build into the model, and improve generalization to unseen data. To the best of our knowledge, this approach has not been systematically explored for music signals. Using the problem of singing voice detection with neural networks as an example, we apply a range of label-preserving audio transformations to assess their utility for music data augmentation. In line with recent research in speech recognition, we find pitch shifting to be the most helpful augmentation method. Combined with time stretching and random frequency filtering, we achieve a reduction in classification error between 10 and 30%, reaching the state of the art on two public datasets. We expect that audio data augmentation would yield significant gains for several other sequence labelling and event detection tasks in music information retrieval.

1. INTRODUCTION

Modern approaches for object recognition in images are closing the gap to human performance [5]. Besides using an architecture tailored towards images (Convolutional Neural Networks, CNNs), large datasets and a lot of computing power, a key ingredient in building these systems is *data augmentation*, the technique of training and/or testing on systematically transformed examples. The transformations are typically chosen to be label-preserving, such that they can be trivially used to extend the training set and encourage the system to become invariant to these transformations. As a complementary measure, at test time, aggregating predictions of a system over transformed inputs increases robustness against transformations the system has not learned to (or not been trained to) be fully invariant to.

While even earliest work on CNNs [13] successfully employs data augmentation, and research on speech recognition – an inspiration for many of the techniques used in

music information retrieval (MIR) – has picked it up as well [9], we could only find anecdotal references to it in the MIR literature [8, 18], but no systematic treatment.

In this work, we devise a range of label-preserving audio transformations and compare their utility for music signals on a benchmark problem. Specifically, we chose the sequence labelling task of singing voice detection: It is well-covered, but best reported accuracies on public datasets are around 90%, suggesting some leeway. Furthermore, it does not require profound musical knowledge to solve, making it an ideal candidate for training a classifier on low-level inputs. This allows observing the effect of data augmentation unaffected by engineered features, and unhindered by doubttable ground truth. For the classifier, we chose CNNs, proven powerful enough to pick up invariances taught by data augmentation in other fields.

The following section will review related work on data augmentation in computer vision, speech recognition and music information retrieval, as well as the state of the art in singing voice detection. Section 3 describes the method we used as our starting point, Section 4 details the augmentation methods we applied on top of it, and Section 5 presents our findings. Finally, Section 6 rounds up and discusses implications of our work.

2. RELATED WORK

For computer vision, a wealth of transformations has been tried and tested: As an early example (1998), Le et al. [13] applied translation, scaling (proportional and disproportional) and horizontal shearing to training images of handwritten digits, improving test error from 0.95% to 0.8%. Krizhevsky et al. [12], in an influential work on large-scale object recognition from natural images, employed translation, horizontal reflection, and color variation. They do not provide a detailed comparison, but note that it allowed to train larger networks and the color variations alone improve accuracy by 1 percent point. Crucially, most methods also apply specific transformations at test time [5].

In 2013, Jaitly and Hinton [9] pioneered the use of label-preserving audio transformations for speech recognition. They find pitch shifting of spectrograms prior to mel filtering at training and test time to reduce phone error rate from 21.6% to 20.5%, and report that scaling mel spectra either in time or frequency dimensions or constructing examples from perturbed LPC coefficients did not help. Concurrently, Kanda et al. [10] showed that combining pitch shift-



© Jan Schlüter and Thomas Grill.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Jan Schlüter and Thomas Grill. “Exploring Data Augmentation for Improved Singing Voice Detection with Neural Networks”, 16th International Society for Music Information Retrieval Conference, 2015.

ing with time stretching and random frequency distortions reduces word errors by 10%, with pitch shifting proving most beneficial and effects of the three distortion methods adding up almost linearly. Cui et al. [3] combined pitch shifting with a method transforming speech to another speaker’s voice in feature space and Ragni et al. [20] combined it with unsupervised training, both targeting uncommon languages with small datasets. To the best of our knowledge, this comprises the full body of work on data augmentation in speech recognition.

In MIR, literature is even more scarce. Li and Chan [18] observed that Mel-Frequency Cepstral Coefficients are sensitive to changes in tempo and key, and show that augmenting the training and/or test data with pitch and tempo transforms slightly improves genre recognition accuracy on the GTZAN dataset. While this is a promising first step, genre classification is a highly ambiguous task with no clear upper bound to compare results to. Humphrey et al. [8] applied pitch shifting to generate additional training examples for chord recognition learned by a CNN. For this task, pitch shifting is not label-preserving, but changes the label in a known way. While test accuracy slightly drops when trained with augmented data, they do observe increased robustness against transposed input.

Current state-of-the-art approaches for singing voice detection build on Recurrent Neural Networks (RNNs). Le-glaive et al. [15] trained a bidirectional RNN on mel spectra preprocessed with a highly tuned harmonic/percussive separation stage. They set the state of the art on the public Jamendo dataset [21], albeit using a “shotgun approach” of training 20 variants and picking the one performing best on the test set. Lehner et al. [16] trained an RNN on a set of five high-level features, some of which were designed specifically for the task. They achieve the second best result on Jamendo and also report results on RWC [4, 19], a second public dataset. For perspective, we will compare our results to both of these approaches.

3. BASE METHOD

As a starting point for our experiments, we design a straightforward system applying CNNs on mel spectrograms.

3.1 Feature Extraction

We subsample and downmix the input signal to 22.05 kHz mono and perform a Short-Time Fourier Transform (STFT) with Hann windows, a frame length of 1024 and hop size of 315 samples (yielding 70 frames per second). We discard the phases and apply a mel filterbank with 80 triangular filters from 27.5 Hz to 8 kHz, then logarithmize the magnitudes (after clipping values below 10^{-7}). Finally, we normalize each mel band to zero mean and unit variance over the training set.

3.2 Network architecture

As is customary, our CNN employs three types of feedforward neural network layers: Convolutional layers convolving a stack of 2D inputs with a set of learned 2D kernels,

pooling layers subsampling a stack of 2D inputs by taking the maximum over small groups of neighboring pixels, and dense layers flattening the input to a vector and applying a dot product with a learned weight matrix.

Specifically, we apply two 3×3 convolutions of 64 and 32 kernels, respectively, followed by 3×3 non-overlapping max-pooling, two more 3×3 convolutions of 128 and 64 kernels, respectively, another 3×3 pooling stage, two dense layers of 256 and 64 units, respectively, and a final dense layer of a single sigmoidal output unit. Each hidden layer is followed by a $y(x) = \max(x/100, x)$ nonlinearity [1].

The architecture is loosely copied from [11], but scaled down as our datasets are orders of magnitude smaller. It was fixed in advance and not optimized further, as the focus of this work lies on data augmentation.

3.3 Training

Our networks are trained on mel spectrogram excerpts of 115 frames (~1.6 sec) paired with a label denoting the presence of voice in the central frame.

Excerpts are formed with a hop size of 1 frame, resulting in a huge number of training examples. However, these are highly redundant: Many excerpts overlap, and excerpts from different positions in the same music piece often feature the same instruments and vocalists in the same key. Thus, instead of iterating over a full dataset, we train the networks for a fixed number of 40,000 weight updates. While some excerpts are only seen once, this visits each song often enough to learn the variation present in the data. Updates are computed with stochastic gradient descent on cross-entropy error using mini-batches of 32 randomly chosen examples, Nesterov momentum of 0.95, and a learning rate of 0.01 scaled by 0.85 every 2000 updates. Weights are initialized from random orthogonal matrices [22].

For regularization, we set the target values to 0.02 and 0.98 instead of 0 and 1. This avoids driving the output layer weights to larger and larger magnitudes while the network attempts to have the sigmoid output reach its asymptotes for training examples it already got correct [14]. We found this to be a more effective measure against overfitting than L2 weight regularization. As a complementary measure, we apply 50% dropout [7] to the inputs of all dense layers.

All parameters were determined in initial experiments by monitoring classification accuracy at optimal threshold on validation data, which proved much more reliable than cross-entropy loss or accuracy at a fixed threshold of 0.5.

4. DATA AUGMENTATION

We devised a range of augmentation methods that can be efficiently implemented to work on spectrograms or mel spectrograms: Two are data-independent, four are specific to audio data and one is specific to binary sequence labelling. All of them can be cheaply applied on-the-fly during training (some before, some after the mel-scaling stage) while collecting excerpts for the next mini-batch, and all of them have a single parameter modifying the effect strength we will vary in our experiments.

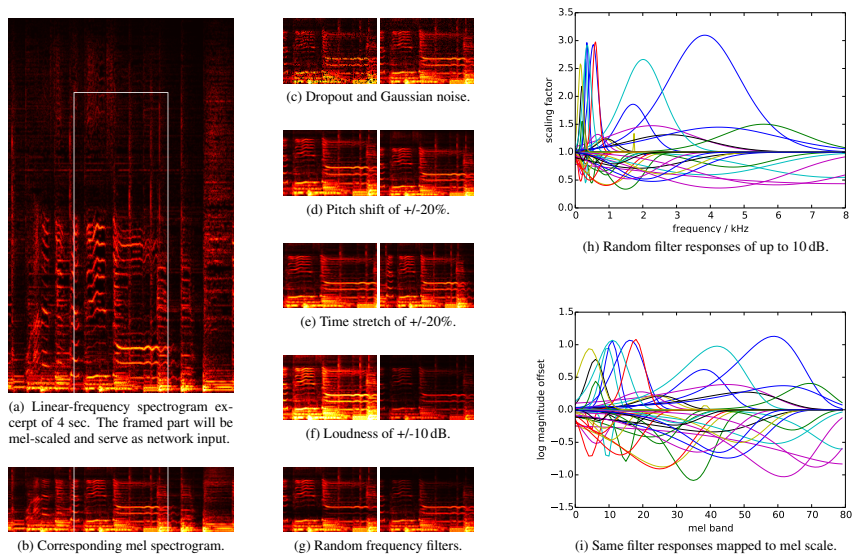


Figure 1: Illustration of data augmentation methods on spectrograms (0:23–0:27 of “Bucle Paranoideal” by LaBarcaDeSua)

4.1 Data-independent Methods

An obvious way to increase a model’s robustness is to corrupt training examples with random noise. We consider **dropout** – setting inputs to zero with a given probability – and additive **Gaussian noise** with a given standard deviation. This is fully independent of the kind of data we have, and we apply it directly to the mel spectrograms fed into the network. Figure 1c shows an example spectrogram excerpt corrupted with 20% dropout and Gaussian noise of $\sigma = 0.2$, respectively.

4.2 Audio-specific Methods

Just like in speech recognition, **pitch shifting** and **time stretching** the audio data by moderate amounts does not change the label for a lot of MIR tasks. We implemented this by scaling linear-frequency spectrogram excerpts vertically (for pitch shifting) or horizontally (for time stretching), then retaining the (fixed-size) bottom central part, so the bottom is always aligned with 0Hz, and the center is always aligned with the label. Finally, the warped and cropped spectrogram excerpt is mel-scaled, normalized and fed to the network. Figure 1a shows a linear spectrogram excerpt along with the cropping borders, and Figures 1d–e show the resulting mel spectrogram excerpt with different amounts of shifting or stretching. During training, the factor for each example is chosen uniformly at random¹ in a given range such as 80% to 120%, and the width of the range defines the effect strength we can vary.

¹ Choosing factors on a logarithmic scale did not improve results.

A much simpler idea focuses on invariance to **loudness**: We scale linear spectrograms by a random factor in a given decibel range, or, equivalently, add a random offset to log-magnitude mel spectrograms (Figure 1f). Effect strength is controlled by the allowed factor (or offset) range.

As a fourth method, we apply random **frequency filters** to the linear spectrogram. Specifically, we create a filter response as a Gaussian function $f(x) = s \cdot \exp(0.5 \cdot (x - \mu)^2 / \sigma^2)$, with μ randomly chosen on a logarithmic scale from 150 Hz to 8 kHz, σ randomly chosen between 5 and 7 semitones, and s randomly chosen in a given range such as -10 dB to 10 dB, the width of the range being varied in our experiments. Figure 1h displays 50 of such filter responses, Figure 1g shows two resulting excerpts. When using this method alone, we map responses to the mel scale, logarithmize them (Figure 1i) and add them to the mel spectrograms to avoid the need for mel-scaling on the fly.

4.3 Task-specific Method

For the detection task considered here, we can easily create additional training examples with known labels by **mixing** two music excerpts together. For simplicity, we only regard the case of blending a given training example A with a randomly chosen negative example B , such that the resulting mix will inherit A ’s label. Mixes are created from linear spectrograms as $C = (1 - f) \cdot A + f \cdot B$, with f chosen uniformly at random between 0 and 0.5, prior to mel-scaling and normalization, but after any other augmentations. We control the effect strength via the probability of the augmentation being applied to any given example.

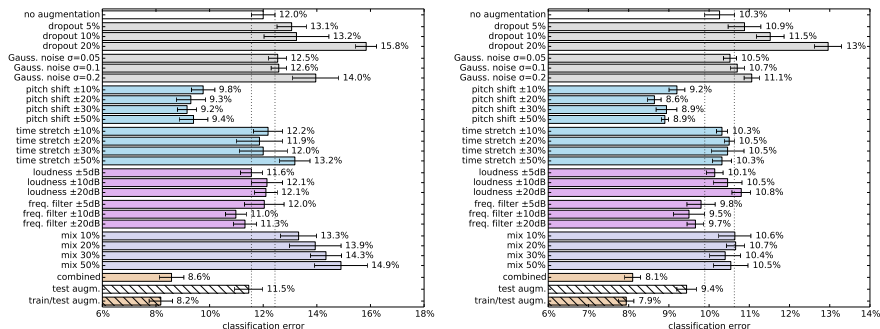


Figure 2: Classification error for different augmentation methods on internal datasets (left: *In-House A*, right: *In-House B*) Bars and whiskers indicate the mean and its 95% confidence interval computed from five repetitions of each experiment.

5. EXPERIMENTAL RESULTS

We first compare the different augmentation methods in isolation at different augmentation strengths on two internal development datasets, to determine how helpful they are and how to parameterize them, and then combine the best methods. In a second set of experiments, we assess the use of augmentation at test time, both for networks trained without and with data augmentation. Finally, we evaluate the best system on two public datasets, comparing against our base system and the state of the art.

5.1 Datasets

In total, we work with four datasets, two of them public:

- *In-House A*: 188 30-second preview snippets from an on-line music store, covering a very wide range of genres and origins. We use 100 files for training, the remaining ones for evaluation.

- *In-House B*: 149 full-length rock songs. While being far less diverse, this dataset features a lot of electric guitars that share characteristics with singing voice. We use 65 files for training, 10 for validation and 74 for testing.

- *Jamendo*: 93 full-length Creative Commons songs collected and annotated by Ramona et al. [21]. For comparison to existing results, we follow the official split of 61 files for training and only 16 files each for validation and testing.

- *RWC*: The RWC-Pop collection by Goto et al. [4] contains 100 pop songs, with singing voice annotations by Mauch et al. [19]. To compare results to Lehner et al. [16], we use the same 5-fold cross-validation split (personal communication).

Each dataset includes annotations indicating the presence of vocals with sub-second granularity. Except for RWC, datasets do not contain duplicate artists.

5.2 Evaluation

At test time, for each spectrogram excerpt, the network outputs a value between 0 and 1 indicating the probability

of voice being present at the center of the excerpt. Feeding maximally overlapping excerpts, we obtain a sequence of 70 predictions per second. Following Lehner et al. [17], we apply a sliding median filter of 800 ms to smoothen the output, then apply a threshold to obtain binary predictions. We compare these predictions to the ground truth labels to obtain the number of true and false positives and negatives, accumulated over all songs in the test set.

While several authors use the F-Score to summarize results, we follow Mauch et al.’s [19] argument that a task with over 50% positive examples is not well-suited for a document retrieval evaluation measure. Instead, we focus on classification error, and also report recall and specificity (recall of the negative class).

5.3 Results on Internal Datasets

In our first set of experiments, we train our network with each of the seven different augmentation methods on each of our two internal datasets, and evaluate it on the (unmodified) test sets. We compare classification errors at the optimal binarization threshold to enable a fair comparison of augmentation methods unaffected by threshold estimation.

Figure 2 depicts our results. The first line gives the result of the base system without any data augmentation. All other lines except for the last three show results with a single data augmentation method at a particular strength.

Corrupting the inputs even with small amounts of noise clearly just diminishes accuracy. Possibly, its regularizing effects [2] only apply to simpler models, as it is not used in recent object recognition systems either [5, 11, 12]. Pitch shifting in a range of $\pm 20\%$ or $\pm 30\%$ gives a significant reduction in classification error of up to 25% relative. It seems to appropriately fill in some gaps in vocal range uncovered by our small training sets. Time stretching does not have a strong effect, indicating that the cues the network picked up are not sensitive to tempo. Similarly, random loudness change does not affect performance. Random frequency filters give a modest improvement, with the

Method	Error	Recall	Spec.
Lehner et al. [16]	10.6%	90.6%	–
Leglaive et al. [15]	8.5%	92.6%	–
Ours w/o augmentation	9.4%	90.8%	90.5%
train augmentation	8.0%	91.4%	92.5%
test augmentation	9.0%	92.0%	90.1%
train/test augmentation	7.7%	90.3%	94.1%

Table 1: Results on Jamendo

Method	Error	Recall	Spec.
Lehner et al. [16]	7.7%	93.4%	–
Ours w/o augmentation	8.2%	92.4%	90.8%
train augmentation	7.4%	93.6%	91.0%
test augmentation	8.2%	93.4%	89.4%
train/test augmentation	7.3%	93.5%	91.6%

Table 2: Results on RWC

best setting at a maximum strength of 10dB. Mixing in negative examples clearly hurts, but a lot less severely on the second dataset. Presumably this is because the second dataset is a lot more homogeneous, and two rock songs mixed together still form a somewhat realistic example, while excerpts randomly mixed from the first dataset are far from anything in the test set. We hoped this would drive the network to recognize voice irrespectively of the background, but apparently this is too hard or besides the task.

The third from last row in Figure 2 shows performance for combining pitch shifting of $\pm 30\%$, time stretching of $\pm 30\%$ and filtering of ± 10 dB. While error reductions do not add up linearly as in [10], we do observe an additional $\sim 6\%$ relative improvement over pitch shifting alone.

5.4 Test-time Augmentation

In object recognition systems, it is customary to also apply a set of augmentations at test time and aggregate predictions over the different variants [5, 11, 12]. Here, we average network predictions (before temporal smoothing and thresholding) over the original input and pitch-shifted input of -20% , -10% , $+10\%$ and $+20\%$. Unsurprisingly, other augmentations were not helpful at test time: Tempo and loudness changes hardly affected training either, and all remaining methods corrupt data.

The last two rows in Figure 2 show results with this measure when training without data augmentation and our chosen combination, respectively. Test-time augmentation is beneficial independently of train-time augmentation, but increases computational costs of doing predictions.

5.5 Final Results on Public Datasets

To set our results in perspective, we evaluate the base system on the two public datasets, adding our combined train-time augmentation, test-time pitch-shifting, or both. For Jamendo, we optimize the classification threshold on the validation set. For RWC, we simply use the optimal threshold determined on the first internal dataset.

As can be seen in Tables 1–2, on both datasets we slightly improve upon the state of the art. This shows that augmentation did not only help because our base system was a weak starting point, but actually managed to raise the bar. We assume that the methods we compared to would also benefit from data augmentation, possibly surpassing ours.

6. DISCUSSION

We evaluated seven label-preserving audio transformations for their utility as data augmentation methods on music data, using singing voice detection as the benchmark task. Results were mixed: Pitch shifting and random frequency filters brought a considerable improvement, time stretching did not change a lot, but did not seem harmful either, loudness changes were ineffective and the remaining methods even reduced accuracy.

The strong influence of augmentation by pitch shifting, both in training and at test-time, indicates that it would be worthwhile to design the classifier to be more robust to pitch shifting in the first place. For example, this could be achieved by using log-frequency spectrograms and inserting a convolutional layer in the end that spans most of the frequency dimension, but still allows filters to be shifted in a limited range.

Frequency filtering as the second best method deserves closer attention. The scheme we devised is just one of many possibilities, and probably far from optimal. A closer investigation of why it helped might lead to more effective schemes. An open question relating to this is whether augmentation methods should generate (a) realistic examples akin to the test data, (b) variations that are missing from the training and test set, but easy to classify by humans, or (c) corrupted versions that rule out inrobust solutions. For example, it is imaginable that narrow-band filters removing frequency components at random would force a classifier to always take all harmonics into account.

Regarding the task of singing voice detection, better solutions would be reached by training larger CNNs or bagging multiple networks, and faster solutions by extracting the knowledge into smaller models [6]. In addition, adding recurrent connections to the hidden layers might help the network to take into account more context in a light-weight way, allowing to reduce the input (and thus, the dense layer) size by a large margin.

Finally, we expect that data augmentation would prove beneficial for a range of other MIR tasks, especially those operating on a low level.

7. ACKNOWLEDGMENTS

This research is funded by the Federal Ministry for Transport, Innovation & Technology (BMVIT) and the Austrian Science Fund (FWF): TRP 307-N23, and the Vienna Science and Technology Fund (WWTF): MA14-018. We also gratefully acknowledge the support of NVIDIA Corporation with the donation of a Tesla K40 GPU used for this research. Last but not least, we thank Bernhard Lehner for fruitful discussions on singing voice detection.

8. REFERENCES

- [1] A. Jannun A. Maas and A. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Int. Conf. on Machine Learning (ICML) Workshop on Deep Learning for Audio, Speech, and Language Processing*, 2013.
- [2] Guozhong An. The effects of adding noise during backpropagation training on a generalization performance. *Neural Comput.*, 8(3):643–674, April 1996.
- [3] Xiaodong Cui, Vaibhava Goel, and Brian Kingsbury. Data Augmentation for Deep Neural Network Acoustic Modeling. In *Proc. of the 2014 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Florence, Italy, 2014.
- [4] Masataka Goto, Hiroki Hashiguchi, Takuichi Nishimura, and Ryuichi Oka. RWC music database: Popular, classical, and jazz music databases. In *Proc. of the 3rd Int. Conf. on Music Information Retrieval (ISMIR)*, pages 287–288, October 2002.
- [5] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. *CoRR*, abs/1502.01852, February 2015.
- [6] G. Hinton, O. Vinyals, and J. Dean. Distilling the Knowledge in a Neural Network. ArXiv:1503.02531, March 2015.
- [7] G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R.R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. arXiv:1207.0580, July 2012.
- [8] Eric J. Humphrey and Juan P. Bello. Rethinking automatic chord recognition with convolutional neural networks. In *Proc. of the 11th Int. Conf. on Machine Learning and Applications (ICMLA)*, 2012.
- [9] Navdeep Jaitly and Geoffrey E. Hinton. Vocal tract length perturbation (VTLP) improves speech recognition. In *Int. Conf. on Machine Learning (ICML) Workshop on Deep Learning for Audio, Speech, and Language Processing*, 2013.
- [10] Naoyuki Kanda, Ryu Takeda, and Yasunari Obuchi. Elastic spectral distortion for low resource speech recognition with deep neural networks. In *Automatic Speech Recognition and Understanding Workshop (ASRU)*, Olomouc, Czech Republic, 2013.
- [11] Karen Simonyan, Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. of the 3rd Int. Conf. on Learning Representations (ICLR)*, May 2015.
- [12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C.J.C. Burges, L. Bottou, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.
- [13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proc. of the IEEE*, 86(11):2278–2324, November 1998.
- [14] Y. LeCun, L. Bottou, G. Orr, and K. Müller. Efficient BackProp. In G. Orr and Müller K., editors, *Neural Networks: Tricks of the trade*. Springer, 1998.
- [15] Simon Leglaive, Romain Hennequin, and Roland Badeau. Singing voice detection with deep recurrent neural networks. In *Proc. of the 2015 IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 121–125, Brisbane, Australia, April 2015.
- [16] Bernhard Lehner, Gerhard Widmer, and Sebastian Böck. A low-latency, real-time-capable singing voice detection method with LSTM recurrent neural networks. In *Proc. of the 23th European Signal Processing Conf. (EUSIPCO)*, Nice, France, 2015.
- [17] Bernhard Lehner, Gerhard Widmer, and Reinhard Sonnleitner. On the reduction of false positives in singing voice detection. In *Proc. of the 2014 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 7530–7534, 2014.
- [18] Tom L.H. Li and Antoni B. Chan. Genre classification and the invariance of MFCC features to key and tempo. In *Proc. of the 17th Int. Conf. on MultiMedia Modeling (MMM)*, Taipei, Taiwan, 2011.
- [19] Matthias Mauch, Hiromasa Fujihara, Kazuyoshi Yoshii, and Masataka Goto. Timbre and melody features for the recognition of vocal activity and instrumental solos in polyphonic music. In *Proc. of the 12th Int. Society for Music Information Retrieval Conf. (ISMIR)*, 2011.
- [20] Anton Ragni, Kate M. Knill, Shakti P. Rath, and Mark J. F. Gales. Data augmentation for low resource languages. In Haizhou Li, Helen M. Meng, Bin Ma, Engsiong Chng, and Lei Xie, editors, *Proc. of the 15th Annual Conf. of the Int. Speech Communication Association (INTERSPEECH)*, pages 810–814, Singapore, 2014. ISCA.
- [21] Mathieu Ramona, Gaël Richard, and Bertrand David. Vocal detection in music with support vector machines. In *Proc. of the 2008 IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1885–1888, 2008.
- [22] Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *Int. Conf. on Learning Representations (ICLR)*, 2014.

AUDIO CHORD RECOGNITION WITH A HYBRID RECURRENT NEURAL NETWORK

Siddharth Sigtia* Nicolas Boulanger-Lewandowski† Simon Dixon*

* Centre for Digital Music, Queen Mary University of London, London, UK

† Dept. IRO, Université de Montréal, Montréal (QC), H3C 3J7, Canada

{s.s.sigtia,s.e.dixon}@qmul.ac.uk

ABSTRACT

In this paper, we present a novel architecture for audio chord estimation using a hybrid recurrent neural network. The architecture replaces hidden Markov models (HMMs) with recurrent neural network (RNN) based language models for modelling temporal dependencies between chords. We demonstrate the ability of feed forward deep neural networks (DNNs) to learn discriminative features directly from a time-frequency representation of the acoustic signal, eliminating the need for a complex feature extraction stage. For the hybrid RNN architecture, inference over the output variables of interest is performed using beam search. In addition to the hybrid model, we propose a modification to beam search using a hash table which yields improved results while reducing memory requirements by an order of magnitude, thus making the proposed model suitable for real-time applications. We evaluate our model's performance on a dataset with publicly available annotations and demonstrate that the performance is comparable to existing state of the art approaches for chord recognition.

1. INTRODUCTION

The ideas presented in this paper are motivated by the recent progress in end-to-end machine learning and neural networks. In the last decade, it has been shown that given a large dataset, deep neural networks (DNNs) are capable of learning useful features for discriminative tasks. This has led complex feature extraction methods to be replaced with neural nets that act directly on raw data or low level features. Current state-of-the-art methods in speech recognition and computer vision employ DNNs for feature extraction [12]. In addition to feature learning, recurrent neural networks (RNNs) have been shown to be very powerful models for temporal sequences [9, 12]. In the field of Music Information Retrieval (MIR), various studies have

applied neural network based models to different tasks [3, 11, 15]. These experiments have been motivated by the fact that hand-crafting features to extract musically relevant information from audio is a difficult task. Existing approaches in MIR would benefit greatly if feature extraction could be automated.

Audio chord recognition is a fundamental problem in MIR (see [13] for a review). At a high level, popular chord recognition algorithms follow a pipeline similar to the one followed in speech. Most systems are comprised of an *acoustic model* which is used to process the acoustic information present in the audio signal. The estimates of the acoustic model are further refined by a *language model* that models the temporal relationships and structure present in sequences of chord symbols. Our proposed approach deviates from existing approaches in two fundamental ways. We use DNNs to learn discriminative features from a time-frequency representation of the audio. This is contrary to the common approach of extracting chroma features (and their many variants) as a preprocessing step. Secondly, we generalise the popular method of using a Hidden Markov Model (HMM) language model with a more powerful RNN based language model. Finally, we combine the acoustic and language models using a hybrid RNN architecture previously used for phoneme recognition and music transcription [5, 14].

In the past, RNNs have been applied to chord recognition and music transcription in a sequence transduction framework [3, 4]. However, these models suffer from an issue known as *teacher forcing*, which occurs due to the discrepancy between the training objective and the way the RNN is used at test time. During training, RNNs are trained to predict the output at any time step, given the correct outputs at all preceding steps. This is in contrast to how they are used at test time, where the RNN is fed predictions from previous time steps as inputs to the model. This can lead to an unsuitable weighting of the acoustic and symbolic information, which can quickly cause errors to accumulate at test time. The hybrid RNN architecture resolves this issue by offering a principled way for explicitly combining acoustic and symbolic predictions [14].

The hybrid RNN model outputs a sequence of conditional probability distributions over the output variables (Section 3). The structure of the graphical model makes the problem of exactly estimating the most likely sequence of outputs intractable. Beam search is a popular heuristic

†NB is currently working at Google Inc., Mountain View, USA



tic graph search algorithm which is used to decode conditional distributions of this form. Beam search when used for decoding temporal sequences is fundamentally limited by the fact that sequences that are quasi-identical (differ at only few time steps) can occupy most of the positions within the beam, thus narrowing the range of possibilities explored by the search algorithm. We propose a modification to the beam search algorithm which we call *hashed beam search* in order to encourage *diversity* in the explored solutions and reduce computational cost.

The rest of the paper is organised as follows: Section 2 describes the feature learning pipeline. Section 3 briefly introduces the hybrid RNN architecture. Section 4 describes the proposed modification to the beam search algorithm. Experimental details are provided in Section 5, results are outlined in Section 6 and the paper is concluded in Section 7.

2. FEATURE LEARNING

We follow a pipeline similar to the one adopted in [3, 15] for feature extraction. We transform the raw audio signal into a time-frequency representation with the constant-Q transform [6]. We first down-sample the audio to 11.025 kHz and compute the CQT with a hop-size of 1024 samples. The CQT is computed over 7 octaves with 24 bins per octave yielding a 168 dimensional vector of real values. One of the advantages of using the CQT is that the representation is low dimensional and linear in pitch. Computing the short-time Fourier transform over long analysis windows would lead to a much higher dimensional representation. Lower dimensional representations are useful when using DNNs since we can train models with fewer parameters, which makes the parameter estimation problem easier.

After extracting CQT frames for each track, we use a DNN to classify each frame to its corresponding chord label. As mentioned earlier, DNNs have been shown to be very powerful classifiers. DNNs learn complex non-linear transformations of the input data through their hidden layers. In our experiments we used DNNs with 3 hidden layers. We constrained all the layers to have the same number of hidden units to simplify the task of searching for good DNN architectures. The DNNs have a softmax output layer and the model parameters are obtained using maximum likelihood estimation.

Once the DNNs are trained, we use the activations of the final hidden layer of the DNN as features. In our experiments we observed that the acoustic model performance was improved ($\sim 3\%$ absolute improvement in frame-level accuracy) if we provided each frame of features with context information. Context information was provided by performing mean and variance pooling over a context window around the central frame of interest [3]. A context window of length $2k + 1$ is comprised of the central frame of interest, along with k frames before and after the central frame. In our experiments we found that a context window of 7 frames provided the best results.

We trained the network with mini-batch stochastic gra-

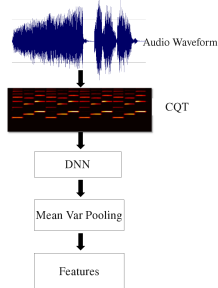


Figure 1. Feature Extraction Pipeline

dent descent. Instead of using learning rate update schedules, we use ADADELTA which adapts the learning rate over iterations [18]. In our experiments we found Dropout was essential to improve generalisation [16]. We found a Dropout rate of 0.3 applied to all layers of the DNN to be optimal for controlling overfitting. Once the models are trained, we use the model that performs best on the validation set to extract features. In our experiments, the best performing model had 100 hidden units in each layer. Figure 1 is a graphical representation of the feature extraction pipeline. In section 6, we compare DNN acoustic models with different feature inputs.

3. HYBRID RECURRENT NEURAL NETWORKS

Similar to language, chord sequences are highly correlated in time. We propose exploiting this structure for audio chord estimation using hybrid RNNs. The hybrid RNN is a generative graphical model that combines the predictions of an arbitrary frame level classifier with the predictions of an RNN language model. For temporal problems, the predictions of the system can be greatly improved by modelling the relationships between outputs, analogous to language modelling in speech. Typically, HMMs are employed in order to model and exploit this structure. Hybrid RNNs generalise the HMM architecture by using powerful RNN language models.

3.1 RNN Language Model

RNNs can be used to define a probability distribution over a sequence $z = \{z_\tau | 0 \leq \tau \leq T\}$ in the following manner:

$$P(z) = \prod_{t=1}^T P(z_t | \mathcal{A}_t) \quad (1)$$

where $\mathcal{A}_t \equiv \{z_\tau | \tau < t\}$ is the sequence history at time t . The above factorisation is achieved by allowing the RNN at time $t - 1$ to predict the outputs at the next time step, yielding the conditional distribution $P(z_t | \mathcal{A}_t)$.

The RNN is able to model temporal relationships via its hidden state which at any time t has recurrent connections

to the hidden state at $t - 1$. The hidden state is updated according to the following equation:

$$h_t = \sigma(W_{zh}z_{t-1} + W_{hh}h_{t-1} + b_h) \quad (2)$$

where W_{zh} are the weights from the inputs at $t - 1$ to the hidden units at t , W_{hh} are the recurrent weights between hidden units at $t - 1$ and t and b_h are the hidden biases. The form of the hidden state (Equation 2) implies that the predictions at time t are explicitly conditioned on the entire sequence history \mathcal{A}_t . This is contrary to HMMs which are constrained by the Markov property. Therefore, theoretically RNNs can model complex and long-term temporal dependencies between outputs. The parameters of the RNN are estimated by using stochastic gradient based methods. Although theoretically very powerful, RNNs are limited by the effectiveness of the optimisation method [2]. The hidden units described above can be replaced by Long-Short Term Memory (LSTM) units in order to improve the parameter estimation and generalisation capabilities of the RNN (see [10] for a review). In our model we use RNNs with LSTM memory units to model the symbolic structure of chord sequences.

3.2 Hybrid Architecture

Given a sequence of acoustic frames x and a sequence of corresponding chord outputs z , the Hybrid RNN model factorises the joint probability of x and z according to the following equation:

$$\begin{aligned} P(z, x) &= P(z_1 \dots z_T, x_1 \dots x_T) \\ &= P(z_1)P(x_1|z_1) \prod_{t=2}^T P(z_t|\mathcal{A}_t)P(x_t|z_t) \\ &\propto P(z_1) \frac{P(z_1|x_1)}{P(z_1)} \prod_{t=2}^T P(z_t|\mathcal{A}_t) \frac{P(z_t|x_t)}{P(z_t)}. \end{aligned} \quad (3)$$

By restricting the acoustic model to operate on an acoustic frame x_t independent of previous inputs and outputs, the distributions $P(z_t|\mathcal{A}_t)$ and $P(z_t|x_t)$ can be independently modelled by an RNN and an arbitrary frame-level classifier, respectively. The form of the joint probability distribution makes maximum likelihood estimation of the model parameters using gradient based optimisers easy. The acoustic and language model terms separate out when optimising the log-likelihood and the model parameters can be trained using gradient based methods according to the following equations where Θ_a, Θ_l are parameters of the acoustic and language models, respectively:

$$\frac{\partial \log P(z, x)}{\partial \Theta_a} = \frac{\partial}{\partial \Theta_a} \sum_{t=1}^T \log P(z_t|x_t) \quad (4)$$

$$\frac{\partial \log P(z, x)}{\partial \Theta_l} = \frac{\partial}{\partial \Theta_l} \sum_{t=2}^T \log P(z_t|\mathcal{A}_t). \quad (5)$$

Although the hybrid RNN has a similar structure (separate acoustic and language models) to the sequence transduction model in [9], the hybrid RNN explicitly combines

the acoustic and language model distributions. The transduction model in [9], models *unaligned* sequences with an implicit exponential duration.

The property that the acoustic and language models can be trained independently has some useful implications. In MIR, it is easier to obtain chord and note transcriptions from the web as compared to audio data due to copyright issues. We can use the abundance of transcribed data to train powerful language models for various tasks, without the need for annotated, aligned audio data.

4. INFERENCE

The hybrid RNN generalises the HMM graph by conditioning z_t on the entire sequence history \mathcal{A}_t , as compared to the HMM graph where z_t is only conditioned on z_{t-1} (Equation 3). This conditioning allows musicological structure learnt by the language model to influence successive predictions. One consequence of the more general graphical structure is that at test time, inference over the output variables at t requires knowledge of all predictions made till time t . At any t , the history \mathcal{A}_t is still uncertain. Making estimates in a greedy chronological manner does not necessarily yield good solutions. Good solutions correspond to sequences that maximise the likelihood globally.

Beam search is a standard search algorithm used to decode the outputs of an RNN [5, 9, 14]. Beam search is a breadth-first graph search algorithm which maintains only the top w solutions at any given time. At time t , the algorithm generates candidate solutions and their likelihoods at $t + 1$, for all the sub-sequences present in the beam. The candidate solutions are then sorted by log-likelihood and the top w solutions are kept for further search. A beam capacity of 1 is equivalent to greedy search and a beam width of N^T is equivalent to an exhaustive search, where N is the number of output symbols and T is the total number of time steps.

Beam search suffers from a pathological condition when used for decoding sequences. Quasi-identical sequences with high likelihoods can saturate the beam. This limits the range of solutions evaluated by the algorithm. This is especially true when decoding long sequences. The performance of beam search can be improved by pruning solutions that are unlikely. The dynamic programming (DP) based pruned beam search algorithm makes better use of the available beam capacity w [3, 5]. The strategy employed for pruning is that at any time t , the most likely sequence with output symbol $z_t \in C$ is considered and other sequences are discarded, where C is the set of output symbols.

Although the DP beam search algorithm performs well in practice [3, 5], pruning based on the last emitted symbol is a strict constraint. In the next section we propose a modification to the beam search algorithm that is more general and allows flexible design to enforce *diversity* in the set of solutions that are explored and to reduce computational cost.

4.1 Hashed Beam Search

As discussed before, beam search can lead to poor estimates of the optimal solution due to saturation of the beam with similar sequences. The efficiency of the search algorithm can be improved by pruning solutions that are sufficiently similar to a sequence with higher likelihood. We propose a more general variant of the pruned beam search algorithm where the metric for similarity of sequences can be chosen according to the given problem. We encode the similarity metric in the form of a *hash function* that determines the similarity of 2 sequences. Given 2 solutions with the same hash value, the solution with the higher likelihood is retained.

The proposed algorithm is more general and flexible since it allows the similarity metric to be chosen based on the particular instance of the decoding problem. We describe the algorithm for decoding chord sequences. We let the hash function be the last n emitted symbols. With this hash function, if there are two candidate solutions with the same sequence of n symbols at the end, then the hash function produces the same key and we retain the solution with the higher likelihood. When $n = 1$, the algorithm is equivalent to the DP beam search algorithm. When $n = \text{len}(\text{sequence})$, then the algorithm is equivalent to regular beam search. Therefore, by increasing the value of n , we can linearly relax the constraint used for pruning in the DP-like beam search algorithm.

Another generalisation that can be achieved with the hash table is that for each hash key, we can maintain a list of k solutions using a process called chaining [17]. This is more general than the DP beam search algorithm where only the top solution is kept for each output symbol. Algorithm 1 describes the proposed hashed beam search algorithm, while Algorithm 2 describes the beam objects. The time complexity of Algorithm 1 is $O(NTw \log w)$. Even though the time complexity of the proposed algorithm is the same as regular beam search, the algorithm is able to significantly improve performance by pruning unlikely solutions (see Section 6). In Algorithms 1 and 2, s is a sub-sequence, l is the log-likelihood of s and f_h is the hash function.

Algorithm 1 Hashed Beam Search

```

Find the most likely sequence  $z$  given  $x$  with a beam width  $w$ .
 $beam \leftarrow$  new beam object
 $beam.insert(0, \{\})$ 
for  $t = 1$  to  $T$  do
   $new\_beam \leftarrow$  new beam object
  for  $(l, s)$  in  $beam$  do
    for  $z$  in  $C$  do
       $l' = \log P_{tm}(z|s)P_{am}(z|x_t) - \log P(z)$ 
       $new\_beam.insert(l + l', \{s, z\})$ 
   $beam \leftarrow new\_beam$ 
return  $beam.pop()$ 

```

Although the description of the proposed algorithm has been within the context of decoding chord sequences, var-

ious other measures of similarity can be constructed depending upon the problem. For example, for chord and speech recognition, we can use the last n unaligned symbols as the hash function (results with chords were uninteresting). For problems where the predictions are obtained from an RNN and frame-based similarity measures are insufficient, we can use a vector quantised version of the final hidden state as the key for the hash table entry.

Algorithm 2 Description of beam objects given w, k, f_h

Initialise beam object

```

 $beam.hashQ =$  dictionary of priority queues*
 $beam.queue =$  indexed priority queue of length  $w$ **

```

Insert l, s into beam

```

 $key = f_h(s)$ 
 $queue = beam.queue$ 
 $hashQ = beam.hashQ[key]$ 
 $fits\_in\_queue = \text{not } queue.full() \text{ or } l \geq queue.min()$ 
 $fits\_in\_hashQ = \text{not } hashQ.full() \text{ or } l \geq hashQ.min()$ 
if  $fits\_in\_queue$  and  $fits\_in\_hashQ$  then
   $hashQ.insert(l, s)$ 
  if  $hashQ.overfull()$  then
     $item = hashQ.del\_min()$ 
     $queue.remove(item)$ 
   $queue.insert(l, s)$ 
  if  $queue.overfull()$  then
     $item = queue.del\_min()$ 
     $beam.hashQ[f_h(item.s)].remove(item)$ 

```

* The dictionary maps hash keys to priority queues of length k which maintain (at most) the top k entries at all times.

** An indexed priority queue allows efficient random access and deletion [1].

5. EXPERIMENTS

5.1 Dataset

Unlike other approaches to chord estimation, our proposed approach aims to learn the audio features, the acoustic model and the language model from the training data. Therefore, maximum likelihood training of the acoustic and language models requires sufficient training data, depending on the complexity of the chosen models. Additionally, we require the raw audio for all the examples in the dataset in order to train the acoustic model which operates on CQTs extracted from the audio. In order to satisfy these constraints, we use the dataset used for the MIREX Audio Chord Estimation Challenge. The MIREX data is comprised of two datasets. The first dataset is the collected Beatles, Queen and Zweieck datasets¹. The second dataset is an abridged version of the Billboard dataset [7].

The Beatles, Queen and Zweieck dataset contains annotations for 217 tracks and the Billboard dataset contains annotations for 740 unique tracks. The corresponding audio for the provided annotations are not publicly available and

¹ <http://www.isophonics.net/>

we had to acquire the audio independently. We were able to collect all the audio for the Beatles, Queen and Zweieck dataset and 650 out of the 740 unique tracks for the Billboard dataset (see footnote² for details), leading to a total of 867 tracks for training and testing³. Although we are not able to directly compare results with MIREX evaluations due to the missing tracks, we show that the training data is sufficient for estimating models of sufficient accuracy and the results are comparable to the top performing entries submitted to MIREX 2014. Keeping in mind the limited number of examples in the dataset, all the ground truth chord annotations were mapped to the major/minor chord dictionary which is comprised of 12 major chords, 12 minor chords and one *no chord* class. All results are reported on 4-fold cross-validation experiments on the entire dataset. For training the acoustic and language models, the *training* data was further divided into a training (80%) and validation split (20%).

5.2 Acoustic Model Training

The features obtained from the DNN feature extraction stage (Section 2) are input to an acoustic model which provides a posterior probability over chord labels, $P(z_t|x_t)$ given an input feature vector. Similar to the feature extraction, we use DNNs with a softmax output layer to model the probabilities of output chord classes. We train models with 3 hidden layers with varying number of hidden units. The acoustic models are trained on a frame-wise basis, independently of the language models. We use stochastic mini-batch gradient descent with ADADELTA for estimating the DNN parameters. We use a constant Dropout rate of 0.3 on all the DNN layers to reduce overfitting. Dropout was found to be essential for good generalisation performance, yielding an absolute performance improvement of up to 4% on the test set. We used a mini-batch size of 100 and early stopping for training. Training was stopped if the log-likelihood of the validation set did not increase for 20 iterations over the entire training set. Unlike the feature extraction stage, we do not discard any of the trained models. Instead of using only the best performing model on the validation set, we average the predictions of all the trained models to form an *ensemble of DNNs* [8] as the acoustic model. We found that simply averaging the predictions of the acoustic classifiers led to an absolute improvement of up to 3% on frame classification accuracies.

5.3 Language Model Training

As outlined in Section 3, we use RNNs with LSTM units for language modelling. The training data for the language models is obtained by sampling the ground truth chord transcriptions at the same frame-rate at which CQTs are extracted from the audio waveforms. We use RNNs with 2 layers of hidden recurrent units (100 LSTM units each) and an output softmax layer. Each training sequence was further divided into sub-sequences of length 100. The RNNs

	Language Model			
	None		LSTM RNN	
	OR	WAOR	OR	WAOR
DNN-CQT	57.0%	56.5%	62.8%	62.0%
DNN-DNN Feats	69.8%	69.1%	73.4%	73.0%
DNN-CW DNN Feats	72.9%	72.5%	75.5%	75.0%

Table 1. 4-fold cross-validation results on the MIREX dataset for the major/minor prediction task. DNN-CQT refers to CQT inputs to a DNN acoustic model. DNN-DNN Feats refers to DNN feature inputs to the DNN acoustic model. DNN-CW DNN Feats refers to DNN features with a context window as input to the acoustic model.

were trained with stochastic gradient descent on individual sub-sequences, without any mini-batching. Unlike the acoustic models, we observed that ADADELTA did not perform very well for RNN training. Instead, we used an initial learning rate of 0.001 that was linearly decreased to 0 over 1000 training iterations. We also found that a constant momentum rate of 0.9 helped training converge faster and yielded better results on the test set. We used early stopping and training was stopped if validation log-likelihood did not increase after 20 epochs. We used gradient clipping when the norm of the gradients was greater than 50 to avoid gradient explosion in the early stages of training.

6. RESULTS

In Table 1, we present 4-fold cross validation results on the combined MIREX dataset at the major/minor chord level. The metrics used for evaluation are the overlap ratio (OR) and the weighted average overlap ratio (WAOR) which are commonly used for evaluating chord recognition systems (including MIREX). The test data is sampled every 10ms similar to the MIREX evaluations. The outputs of the hybrid model were decoded with the proposed hashed beam search algorithm. A grid search was performed over the decoding parameters and the presented results correspond to the parameters that were determined to be optimal over the training set.

From Table 1, it is clear that the hybrid model improves performance over the acoustic-only models. The results show that the performance of the acoustic model is greatly improved when the input features to the model are learnt by a DNN as opposed to CQT inputs. The performance of the acoustic model is further improved (3% absolute improvement) when mean and variance pooling is performed over a context window of DNN features. It is interesting to note that the relative improvement in performance is highest for the DNN-CQT and DNN-DNN Feats configurations. This is due to the fact that the hybrid model is derived with the explicit assumption that given a state z_t , the acoustic frame x_t is conditionally independent of all state and acoustic vectors occurring at all other times. Applying a context window to the features violates this independence assumption and therefore the relative improvement is diminished. However, the improved performance of the acoustic model

² www.cecs.qmul.ac.uk/~sss31

³ AUDFPRINT was used to align the audio to the corresponding annotations: <http://labrosa.ee.columbia.edu/matlab/audfprint/>

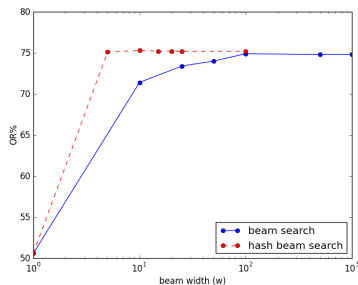


Figure 2. Effect of varying beam width on OR on MIREX data. $n = 2, k = 1$

due to context windowing offsets this loss. Although formal comparisons cannot be made, the accuracies achieved with the hybrid model are similar to the best performing model submitted to the 2014 MIREX evaluation⁴ on the Billboard dataset (OR = 75.57%).

To investigate the advantage of the proposed hashed beam search algorithm, we plot the overlap ratio against the beam width. Figure 2 illustrates that the proposed algorithm can achieve marginally better decoding performance at a significant reduction in beam size. As an example, the hashed beam search yields an OR of 75.1% with a beam width of 5, while regular beam search yields 74.7% accuracy with a beam width of 1000. The time taken to run the hash beam search ($w = 5$) over the test set was 5 minutes, as compared to the regular beam algorithm ($w = 1000$) which took 17 hours to decode the test set. The algorithm’s ability to yield good performance at significantly smaller beam widths indicates that it performs efficient pruning of similar paths, thus utilising the available beam width more efficiently. The run-times of the algorithm show that it can be used for real-time applications without compromising recognition accuracy.

In addition to the beam width, the hash beam search algorithm allows the user to specify the similarity metric and the number of solutions for each hash table entry. We investigate the effect of these parameters on the OR and plot the results in Figure 3. We let the similarity metric be the previous n frames and observe performance as n is linearly increased for a fixed beam width of 25. From Figure 3 we observe that the performance is quite robust to changes in the number of past frames for small values of n . One possible explanation for the graph is that since the test data is sampled at a frame rate of 10ms, all occurrences of chords last for several frames. Therefore counting the previous n frames, effectively leads to the same metric each time. We experimented with using the previous n unique frames as a metric but found that the results deteriorated quite drastically as n was increased. This might reflect the limited

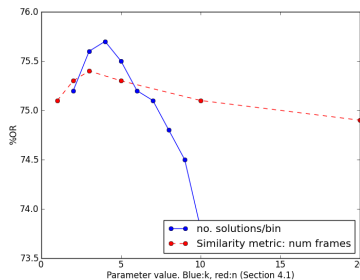


Figure 3. Effect of varying hashed beam search parameters f_h, k on OR on MIREX dataset. $w = 25$.

memory of RNN language models and the issues caused due to lack of explicit duration modelling. The blue line in Figure 3 illustrates the effect of varying the number of solutions per hash table entry. From this graph we see that performance deteriorates significantly once the number of entries per bin crosses a certain threshold (~ 5). This is due to the fact that maintaining many solutions of the same kind saturates the beam capacity with very similar solutions, limiting the breadth of search.

7. CONCLUSION AND FUTURE WORK

We present a chord estimation system based on a hybrid recurrent neural network and the results are competitive with existing state-of-the-art approaches. We show that DNNs are powerful acoustic models. By learning features, they eliminate the need for complex feature engineering. The hybrid RNN model allows us to superimpose an RNN language model on the acoustic model predictions. Additionally, language models can be trained on chord data from the web without the corresponding audio. The results clearly indicate that the language model helps improve model performance by modelling the temporal relationships between output chord symbols and refining the predictions of the acoustic model. The proposed variant of the beam search algorithm significantly reduces memory usage and run times, making the model suitable for real-time applications.

In the future, we would like to conduct chord recognition experiments on larger datasets. This is because the modelling and generalisation capabilities of neural networks improve with more available data for training. An important issue that remains with respect to RNN language models is the problem of duration modelling. Although RNNs are very good at modelling the transition probabilities between events, durations of each event are not modeled explicitly. For musical applications like chord recognition and music transcription, accurate estimates for durations of note occurrences can further help improve the effectiveness of RNN based language models.

⁴ www.music-ir.org/mirex/wiki/2014:Audio.Chord.Estimation.Results

8. REFERENCES

- [1] <https://pypi.python.org/pypi/pqdict/>.
- [2] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, 5(2):157–166, 1994.
- [3] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Audio Chord Recognition with Recurrent Neural Networks. In *The International Society for Music Information Retrieval Conference (ISMIR)*, pages 335–340, 2013.
- [4] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. High-dimensional sequence transduction. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3178–3182. IEEE, 2013.
- [5] Nicolas Boulanger-Lewandowski, Jasha Droppo, Mike Seltzer, and Dong Yu. Phone sequence modeling with recurrent neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5417–5421. IEEE, 2014.
- [6] Judith C Brown. Calculation of a constant Q spectral transform. *The Journal of the Acoustical Society of America*, 89(1):425–434, 1991.
- [7] John Ashley Burgoyne, Jonathan Wild, and Ichiro Fujinaga. An Expert Ground Truth Set for Audio Chord Recognition and Music Analysis. In *The International Society for Music Information Retrieval Conference (ISMIR)*, pages 633–638, 2011.
- [8] Thomas G Dietterich. Ensemble methods in machine learning. In *Multiple Classifier Systems*, pages 1–15. Springer, 2000.
- [9] Alex Graves. Sequence transduction with recurrent neural networks. In *Representation Learning Workshop, International Conference on Machine Learning (ICML)*, 2012.
- [10] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. LSTM: A Search Space Odyssey. *arXiv preprint arXiv:1503.04069*, 2015.
- [11] Eric J Humphrey and Juan Pablo Bello. Rethinking automatic chord recognition with convolutional neural networks. In *11th International Conference on Machine Learning and Applications (ICMLA)*, volume 2, pages 357–362. IEEE, 2012.
- [12] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.
- [13] Matt McVicar, Raúl Santos-Rodríguez, Yizhao Ni, and Tjil De Bie. Automatic chord estimation from audio: A review of the state of the art. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(2):556–575, 2014.
- [14] S. Sigtia, E. Benetos, N. Boulanger-Lewandowski, T. Weyde, Artur S. d’Avila Garcez, and S. Dixon. A Hybrid Recurrent Neural Network for Music Transcription. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 2061–2065, Brisbane, Australia, April 2015.
- [15] Siddharth Sigtia and Simon Dixon. Improved music feature learning with deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6959–6963. IEEE, 2014.
- [16] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.
- [17] Clifford Stein, T Cormen, R Rivest, and C Leiserson. *Introduction to algorithms*, volume 3. MIT Press Cambridge, MA, 2001.
- [18] Matthew D Zeiler. ADADELTA: An adaptive learning rate method. *arXiv preprint arXiv:1212.5701*, 2012.

DESIGN AND EVALUATION OF A PROBABILISTIC MUSIC PROJECTION INTERFACE

Beatrix Vad¹ Daniel Boland¹ John Williamson¹
Roderick Murray-Smith¹ Peter Berg Steffensen²

¹ School of Computing Science, University of Glasgow, United Kingdom

² Syntonetic A/S, Copenhagen, Denmark

mail@bea-vad.de, {daniel, jhw, rod}@dcs.gla.ac.uk, pbs@syntonetic.com

ABSTRACT

We describe the design and evaluation of a probabilistic interface for music exploration and casual playlist generation. Predicted subjective features, such as mood and genre, inferred from low-level audio features create a 34-dimensional feature space. We use a nonlinear dimensionality reduction algorithm to create 2D music maps of tracks, and augment these with visualisations of probabilistic mappings of selected features and their uncertainty.

We evaluated the system in a longitudinal trial in users' homes over several weeks. Users said they had fun with the interface and liked the casual nature of the playlist generation. Users preferred to generate playlists from a local neighbourhood of the map, rather than from a trajectory, using neighbourhood selection more than three times more often than path selection. Probabilistic highlighting of subjective features led to more focused exploration in mouse activity logs, and 6 of 8 users said they preferred the probabilistic highlighting mode.

1. INTRODUCTION

To perform information retrieval on music, we typically rely on either meta data or on 'intelligent' signal processing of the content. These approaches create huge feature vectors and as the feature space expands it becomes harder to interact with. A projection-based interface can provide an overview over the collection as a whole, while showing detailed information about individual items in context. Our aim is to build an interactive music exploration tool, which offers interaction at a range of levels of engagement, which can foster directed exploration of music spaces, casual selection and serendipitous playback. It should provide a consistent, understandable and salient layout of music in which users can learn music locations, select music and generate playlists. It should promote (re-)discovery of music and accommodate widely varying collections.



© Beatrix Vad, Daniel Boland, John Williamson, Roderick Murray-Smith, Peter Berg Steffensen. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Beatrix Vad, Daniel Boland, John Williamson, Roderick Murray-Smith, Peter Berg Steffensen. "Design and evaluation of a probabilistic music projection interface", 16th International Society for Music Information Retrieval Conference, 2015.

To address these goals we built and evaluated a system to interact with 2D music maps, based on dimensionally-reduced inferred subjective aspects such as mood and genre. This is achieved using a flexible pipeline of acoustic feature extraction, nonlinear dimensionality reduction and probabilistic feature mapping. The features are generated by the commercial Moodagent Profiling Service¹ for each song, computed automatically from low-level acoustic features, based on a machine-learning system which learns feature ratings from a small training set of human subjective classifications. These inferred features are uncertain. Subgenres of e.g. electronic music are hard for expert humans to distinguish, and even more so for an algorithm using low-level features [24]. This motivates representing the uncertainty of features in the interaction.

It is not straightforward to evaluate systems based on interacting with such high-dimensional data. This is not a pure visualisation task. Promoting understanding is secondary to offering a compelling user experience, where the user has a sense of control. How do we evaluate projections, especially if the user's success criterion is just to play something 'good enough' with minimal effort? We evaluated our system to answer:

1. Can a single interface enable casual, implicit and focused interaction for music retrieval?
2. Which interface features better enable people to navigate and explore large music collections?
3. Can users create viable mental models of a high-dimensional music space via a 2D map?

2. BACKGROUND

2.1 Arranging music collections on fixed dimensions

A music retrieval interface based on a 2D scatter plot with one axis ranging from slow to fast and the other from dark to bright on the timbre dimension is presented in [10]. The authors show this visualisation reduces time to select suitable tracks compared to a traditional list view. [11] presents a 2D display of music based on the established arousal-valence (AV) diagram of emotions [20], with AV judgements obtained from user ratings. An online exploration tool musiccovery.com [6] enables users to select a mood in the AV space and starts a radio stream based

¹ <http://www.moodagent.com/>

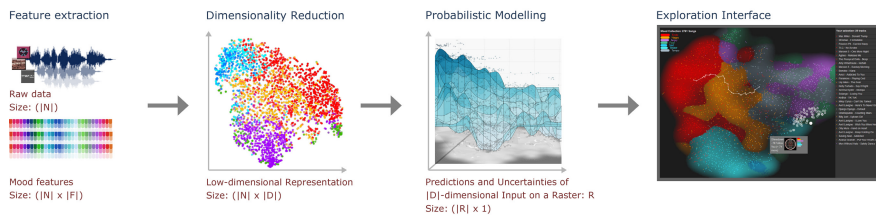


Figure 1. (a) An audio collection, described by a large set of features automatically extracted from the content. (b) visualisation of this high-dimensional dataset in two dimensions using dimensionality reduction (c) probabilistic models showing the distribution of specific features in the low dimensional space (d) combining dimensionality reduction with these models to build an interactive exploration interface.

on the input. These use two predefined dimensions that are easy to interpret, however they do not allow a broader interpretation of musical characteristics based on richer feature sets. [13] finds that music listening is often based upon mood. The investigation of musical preferences in [9] shows most private collections consist of a wide range of styles and approaches to categorisation.

2.2 Music visualisations via dimensionality reduction

“Islands of Music” [17] visualises music collections using a landscape metaphor. They use rhythmic patterns in a set of frequency bands to create a Self-Organizing Map (SOM), a map of music for users to explore. Similarly, [16] introduce the SOM-based PlaySOM and PocketSOM interfaces. Features are again based on rhythm and 2D embedding. An interesting visualisation feature is the use of “gradient fields” to illustrate the distribution of features over the map. Playlist generation is enabled with a rectangular marquee and path selection. Elevations are based on the density of songs in the locality, so clustered songs form islands with mountains. A collection of 359 pieces was used to evaluate the system and song similarities were subjectively evaluated. An immersive 3D environment for music exploration, again using a SOM is described in [14]. An addition to previous approaches is an integrated feedback loop that allows users to reposition songs, alter the terrain and position landmarks. The users’ sense of similarity is modelled and the map gradually adapted. Both the SOM landscape and acoustic clues improved search times per song.

SongWords [2] is an interactive tabletop application to browse music based on lyrics. It combines a SOM with a zoomable user interface. The app is evaluated in a user study with personal music collections of ca. 1000 items. One reported issue was that only the item positions described the map’s distribution of characteristics. Users had to infer the structure of the space from individual items. “Rush 2” explores interaction styles from manual to automatic [1]. They use similarity measures to create playlists automatically by selecting a seed song.

A detailed overview of music visualisation approaches and the MusicGalaxy system is contributed with [23]. This

work introduces adaptive methods for music visualisation, allowing users to adjust weightings in the projection. It also explores the use of a lens so that users could zoom into parts of the music space. Most notably, it receives a significant amount of user evaluation. The lack of such evaluations in the field of MIR has been noted in [21], which calls for a user-centred approach to MIR. The work in this paper thus includes an ‘in-the-wild’ longitudinal evaluation, bringing HCI methodology to bear in MIR.

2.3 Interaction with music visualisations

Path drawings on a music visualisation, enabling high-level control over songs and progression of created playlists can be found in [26]. Casual interaction has recently started receiving attention from the HCI community [18], outlining how interactions can occur at varying levels of engagement. A radio-like interface that adapts to user engagement is introduced by [3, 4]. It allows users to interact with a stream of music at varying levels of control, from casual mood-setting to engaged interaction. Music visualisations can also span engagement – from broad selections in an overview to specific zoomed-in selections.

3. PROBABILISTIC MUSIC INTERFACE

As shown in Figure 1, the interface builds on features derived from raw acoustic characteristics and transforms these into a mood-based visualisation, where nearby songs will have a similar subjective “feeling”. Our feature extraction service provides over thirty predicted subjective features for each song including mood, genre, style, vocals, instrument, beat, tempo, energy and other attributes. The features associated with moods chosen for highlighting in the visualisation include *Happy*, *Angry*, *Sad* and *Tender*. These were identified as relevant moods from social tags in [12]. *Erotic*, *Fear* and *Tempo* (not strictly a mood) were also included. The features were investigated in [5].

Given our large number of features, we need dimensionality reduction to compress the data from $|F|$ dimensions to $|D|$ dimensions. The goal of this step is to preserve subjective similarities between songs and maintain coherent structure in the dataset. For interaction, we reduce down

to 2D. We tried our system with a number of dimensionality reduction techniques including PCA and SOM. We chose the t-distributed stochastic neighbour embedding (t-SNE, [25]) model for non-linear dimensionality reduction to generate a map entangling a global overview of clusters of similar songs and yet locally minimise false positives.

To provide additional information about the composition of the low-dimensional space, we developed *probabilistic models* to visualise high dimensional features in the low-dimensional space. This probabilistic back-projection gives users insight into the structure of the layout, but also into the uncertainties associated with the classifications. On top of the pipeline (Figure 1), we built an efficient, scalable web-based UI which can handle music collections upwards of 20000 songs. The tracks can be seen as random variables drawn from a probabilistic distribution with respect to a specific feature. The distribution parameters can be estimated and used for prediction, allowing smoothed interpolation of features as shown in Figure 2. We used Gaussian Process (GP) priors [19], a powerful nonparametric Bayesian regression method. We applied a squared exponential covariance function on the 2D (x, y) coordinates, predicting the mood features P_f over the map. The GP can also infer the uncertainty σ_f^2 of the predicted feature relevance for each point [22].

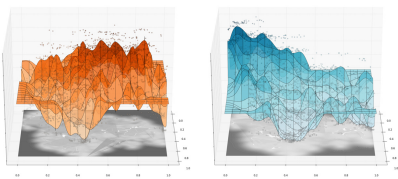


Figure 2. Gaussian Process predictions of features. Orange denotes the “happy” feature distribution and blue denotes “tender”. The greyscale surface shows the uncertainty; lighter is more certain and darker is less certain.

3.1 Interface design

To present the inferred subjective results to the users, the GP mean and standard deviation is evaluated over a 200×200 grid covering the 2D music space. A continuously coloured *background highlighting* is created where areas of high feature scores stand out above areas with higher uncertainty or lower scores. To highlight areas with high prediction scores and low uncertainty, a non-linear transform is used: $\alpha_f = P_f^2 - \sigma_f^2$, for each mood feature f , having a standard deviation σ_f and a predicted feature value P_f . The clusters in the music space can be emphasised as in the upper part of Figure 3 by colouring areas with the colour associated with the highest score; i.e. $\text{argmax}(\alpha_f)$ – a winner-takes-all view. This not only divides the space into discrete mood areas but also shows nuanced gradients of mood influences within those areas. However, once a user starts to dynamically explore a specific area of the space, the system transitions to *implicit*

background highlighting such that the background distribution of the mood with the highest value near the cursor is blended in dynamically as in the lower plots of Figure 3, giving the user more subtle insights into the nature of the space.

Tracks are represented as circles in a scatter plot, where size can convey information, e.g. the popularity of a song, without disturbing the spatial layout. To support visual clustering, colour highlights the highest scoring mood feature of each song, and transparency conveys the feature score. However, the number of diverging, bright colours for categorisation is limited. Murch [15] states that a “refocus” is needed to perceive different pure colours, so matched pairs of bright and desaturated colours are chosen for the correlated pairs *tender/sad*, *happy/erotic* and *angry/fear*.

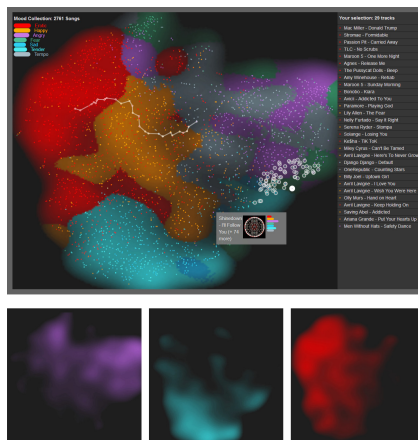


Figure 3. Top: The interactive web interface in its ‘winner takes all’ overview colouring. A path playlist selection as well as a neighbourhood selection is visible in the mood space. Bottom: Background highlighting for the features *angry*, *tender* and *erotic*. Compared with the overview colouring, the subtle fluctuations of features are apparent.

3.2 Interaction with the interface

As the visualisation can handle very large numbers of items, a *semantic zoom* was integrated, where the size of each element is fixed. This coalesces items on zoom out and disentangles items on zoom in.

Further insight into the nature of the space is given by the *adaptive area exploration* tool which visualises the local item density. In contrast to previous work we do not use a fixed selection area but one based on the k -nearest-neighbours to the mouse cursor. Points are highlighted as the mouse is moved, creating a dynamically expanding and collapsing highlight, responding to the structure of the space. The k -nn visualisation adapts to zoom level; when

zoomed out, k is large; when zoomed in, we support focused interaction, with a smaller k .

Focus and context: To make the music map exploration more concrete, a hover box is displayed with information about the nearest item to the cursor, including artist name, title, and album art (see Figure 3). It shows a mini bar chart of the song’s mood features. As this is fixed onscreen, users can explore and observe changes in the mood chart, giving them insight into the high-dimensional space.

3.3 Playlist generation

Neighbourhood selection is a quick and casual interaction metaphor for creating a playlist from the k nearest neighbours. Songs are ranked according to their query point distance. This enables the directed selection of clusters in the space, even if the cluster is asymmetric. By adjusting zoom level (and thus k), k -NN selection can include all in-cluster items while omitting items separated from the perceived cluster. This feature could be enhanced by adding an ϵ -environment similar to the density-based clustering algorithm DBSCAN [7]. Fast rendering and NN search was implemented using quadtree spatial indexing [8].

Path selection enables space-spanning selections. Drawing a path creates a playlist which ‘sticks’ to nearby items along the way. The local density of items is controlled by modulating velocity, so faster trajectory sections stick to fewer songs than slow ones. This ‘dynamic attachment’ offers control over the composition of playlists without visual clutter. E.g. a user can create a playlist starting in the *happy* area, then gradually migrating towards *tender*.

4. USER EVALUATION

The evaluation was based on the *research questions*:

1. How do users perceive the low-dimensional mood space projection?
2. Is the mood-based visualisation useful in music exploration and selection?
3. Which techniques do users develop to create playlists?

A pilot study evaluated the viability of the system and guided the design of the main longitudinal “in the wild” user study, which was conducted to extract detailed usage behaviour over the course of several weeks. Adapting to a new media interface involves understanding how personal preferences and personal media collections are represented. Longitudinal study is essential for capturing the behaviour that develops over time, beyond superficial aesthetic reactions and can – in contrast to Lab-based study – cover common use cases (choose tracks for a party, play something in the background while studying).

Eight participants (1 female, 7 male, 5 from the UK and 3 from Germany, undergraduate and research students) – each with their own Spotify account and personal music collections – were recruited. The mood interface was used to visualise the personal music collection of the participants. The participants used the interface at home as their music player to whatever extent and in whatever way they wanted. Two participants also used the system at work. All subjects used a desktop to access the interface. As a

reward and to facilitate use together with the Spotify Web Player, participants were given a voucher for a three month premium subscription of Spotify.

The Shannon entropy H of the 6 mood features of each user’s music collection gives an impression of the diversity of content. Using the maximum mood feature for each song, $H = -\sum_i p_i \log_2 p_i$, where $p_i = N_{mood_i}/N$.

	1	2	3	4	5	6	7	8
H	2.51	2.36	2.49	2.42	1.84	2.38	1.93	2.5
N	3679	2623	4218	3656	2738	2205	1577	3781

Table 1. Entropy H , no. tracks N of users’ collections.

The study took place in two blocks, each with nominally four days of usage, although the actual duration varied slightly. One of the key aims was to find out if the probabilistic background highlighting provides an enhanced experience, so the study was comprised of two conditions in a counterbalanced within-subjects arrangement:

A Music Map without background highlighting.

B Music Map with background highlighting: The probabilistic models are included, with the composite view of the mood distribution as well as dynamic mood highlighting on cursor movements. Each participant was randomly assigned either condition **A** in week 1 followed by **B** in week 2 or vice versa. At the beginning of each condition and the end of the study, questionnaires were administered to capture participants’ experience with the interface. Interface events, including playlist generation, navigation and all mouse events (incl. movements) were recorded.

5. RESULTS

Most participants used the software extensively, generating an average of 21 playlists per user per week, as shown in Table 2. On average, users actively interacted with the system for 77 minutes each week (roughly 20 minutes a day) – time spent passively listening to playlists is not included in this figure. Both groups generated more playlists in week 1 than in week 2, as they explored the system.

User	1	2	3	4	5	6	7	8
$N_{p,A}$	27	164	4	7	18	8	5	25
$N_{p,B}$	46	39	3	7	5	17	18	53

Table 2. No. playlists generated per user for cond. A & B. Users 1-5 had A in week 1, while 6-8 had A in week 2.

5.1 Mood perception

After each condition, users were asked to rate their satisfaction with interacting via the mood space. The overall opinion was encouraging. The majority of participants reported that they felt their collection was ordered in a meaningful way. Six stated that the mood-based categorisation made sense. Initially, the distinction of different music types was not rated as consistently over all conditions. This might be due to the fact that people usually discuss music in terms of genres rather than moods. However, the difficulty rating of mood changed over sessions. While six users rated mood-based categorisation as difficult at the

start, only three participants still rated mood as difficult to categorise by the second week. This suggests that users can quickly learn the unfamiliar mood-based model.

5.2 Interactions with the Mood Space

Browsing the Space: Analysis of mouse movements provided insight into how participants explored mood spaces. Heatmaps were generated showing the accumulated mouse positions in each condition (Figure 4). Participants explored the space more thoroughly in week one of the study. Some participants concentrated exploration on small pockets, while others explored the whole space relatively evenly.

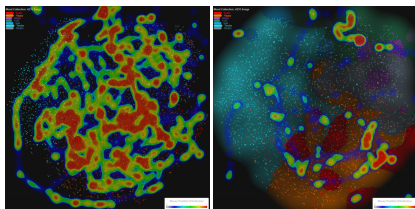


Figure 4. Heatmaps of interaction (mouse activity) of user 3 in week 1 (left) and week 2 (right). Interaction becomes more focused in the second week.

The browse/select ratio dropped noticeably for the second week for users with condition **A** first, as shown in Table 3. This suggests that participants browsed much more for each playlist in the first part of the study, and were more targeted in the second part. The browsing could have been either curiosity-driven exploration, or a frustrating experience, because the non-linear nature of the mapping made the space difficult for the users to predict the response to movements in the space. However, from Table 3 we can see that users who had the highlights in the first week seemed to have much more focused navigation from the start, and did not decrease their browsing much in week 2 when they lost the highlighting mechanism.

	Condition A	Condition B
Week 1	1218.39 (483.49)	447.94 (176.85)
Week 2	319.25 (29.27)	576.5 (416.72)

Table 3. Browse select ratios (std. dev. in parentheses) for week 1 and week 2 of the experiment, in cond. A and B.

Selections and Playlist Creation: Figure 5 shows playlists from two different participants in condition **B**. User 1 (left) created playlists by neighbourhood selection, and also drew a few trajectory playlists. User 7 (right) moved in over a more diverse set with a number of trajectory playlists. The paths partially follow the contours of the background highlight, which suggests this user explored contrasts in the mood space on these boundaries.

Neighbourhood selection was used more often (341 neighbourhood selections and 105 path selections). A rise

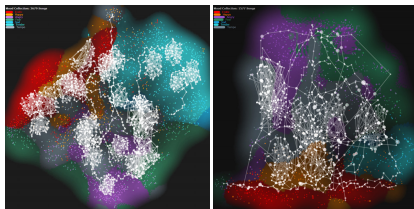


Figure 5. Created playlists under condition **B** for user 1 ($H = 2.51$) and user 7 ($H = 1.93$). Note the different class layouts for the collections with high/low entropy H .

in the use of path selections, and a decline in neighbourhood selections can be seen in condition **B** versus **A**. In condition **A**, five times more neighbourhood than path selections were recorded, and only twice as many in condition **B** (see Table 4). This could be explained by the background distributions suggesting mood influence change gradually over the space. This information may encourage users to create trajectory playlists that gradually change from one mood intensity to another.

Selection	A	B	Total
Path	42	63	105
Neighbourhood	216	125	341
Neighbourhood/Path	5.1×	2.0×	3.3×

Table 4. Usage of the two different selection types in each condition. The neighbourhood/path ratio shows the increased use of the path tool in condition **B**.

5.3 Qualitative feedback

Background Highlighting: We asked whether background highlighting was valuable to the users. The answer was clearly in favour of background highlighting: 6/8 users valued the highlighting, one user was indifferent and one preferred the version without highlighting. The reasons given in favour of the highlighting were that they could more easily identify different regions and remember specific “locales” in the mood space. They recognised that songs had different mood influences and enjoyed following the colour highlights to areas of different intensity. One user stated that he liked the vividness of the implicit highlighting. The user who preferred no highlighting found it a cleaner look that was less confusing. 6 participants stated that they did not find the highlighting confusing. 7 participants answered that it did not distract from the playlist creation task. Qualitative feedback also indicated a preference for highlighting: “[with highlighting] I could easier identify how the mood was distributed over my library”, “coloured areas provided some kind of ‘map’ and ‘landmarks’ in the galaxy”.

Preference for neighbourhood versus path playlists: The domination of neighbourhood versus path playlists in the logged data is supported by feedback from questionnaires,

which shows that users were generally happier with neighbourhood selection than the more novel path selection technique. The attitude towards the path selection differed between conditions. Participants were more satisfied with path selection in condition **B**, with interactive background highlighting. In **A**, four participants agreed the path playlist was effective, and three disagreed. After condition **B**, however, 5 users agreed and only one disagreed.

Advantages of the Interface: The subjective feedback revealed that users had fun exploring the mood space and enjoyed the casual creation of playlists. *"fun to explore the galaxy", "easy generation of decent playlists"* Users also appreciated the casual nature of the interface: *"It was very easy to take a hands-off approach", "I didn't have to think about specific songs"*. Users made specific observations indicating that they were engaged in the exploration task and learned the structure of the map, although this varied among users. *"I discovered that most older jazz pieces were clustered in the uppermost corner", "It was easy to memorize such findings [...] the galaxy thus became a more and more personal space"*. Satisfaction with the quality of selections was high, although some participants found stray tracks that did not fit with neighbouring songs. *"The detected mood was a bit off for a few songs"*. Several users stated that they appreciated the consistency of created playlists and the diversity of different artists, in contrast to their usual artist-based listening. There was concern that playlists did not offer enough diversity *"some songs that dominated the playlists", "too much weight given to 'archive' material", "some way to reorder the playlists to keep them fresh"*, while others enjoyed this aspect: *"I rediscovered many songs I had not listened to in a long time"*.

Shared versus personal: Visualising a shared (i.e. inter-user) mood space with personal collections embedded was not rated very important by most users (only one user thought this important). However, personalisation of the space was rated of high importance by half of the users. Ensuring that nearby songs are subjectively similar was additionally rated as important by the majority of participants (five users). These user priorities led to trade-offs between very large music maps and maps reliably uncovering intrinsic clusters of similar items.

Improvement requests: The most requested missing feature was a text search feature. The use of Spotify for playback also led to a disjointed user experience which would be easily improved on in a fully integrated mood-map music player. Users also requested the integration of recommendations and the ability to compare different mood spaces.

6. CONCLUSIONS AND FUTURE WORK

We presented an interactive tool for music exploration, with musical mood and genre inferred directly from tracks. It features probabilistic representations of multivariable predictions of subjective characteristics of the music to give users subtle, nuanced visualisations of the map. These explicitly represent the vagueness and overlap among fea-

tures. The user-based, in-the-wild evaluation of this novel highlighting technique provided answers to the initial research questions:

Can users create viable mental models of the music space?

The feedback from the 'in-the-wild' evaluation indicates that people enjoyed using these novel interfaces on their own collections, at home, and that mood-based categorisation can usefully describe personal collections, even if initially unfamiliar. Analysis of logged data revealed distinct strategies in experiencing the mood space. Some users explored diverse parts of the mood space and switched among them, while others quickly homed in on areas of interest and then concentrated on those. The questionnaire responses suggest they learned the composition of the space and used it more constructively in the later sessions. Users made plausible mental models of the visualisation – they know where the favourite songs are – and can use this model to discover music and formulate playlists.

Which interface features enable people to navigate and explore the music space?

Interactive background highlighting seemed to reduce the need to browse intensively with the mouse (Table 3). Subjective feedback confirmed that it helped understand the music space with 6/8 users preferring it over no highlighting. Most users did not feel disturbed by the implicitly changing background highlighting. Both the neighbourhood and path playlist generators were used by the participants, although neighbourhood selections were subjectively preferred and were made three times more often than path selections. Subjective feedback highlights the contrast between interfaces which adapt to an individual user taste or reflect a global model, in which all users can collaborate, share and discuss music, trading greater relevance versus greater communicability. Similarly, how can we adapt individual user maps as the user's musical horizons are expanded via the exploratory interface? Users' preference of comparing visualisations over interacting in one large music space hints that an alignment of visualisations is a valid solution to this problem.

Can a single interface enable casual, implicit and focused interaction?

Users valued the ability to vary the level of engagement. Their feedback also suggested that incorporating preview and control over the playing time of playlists would be useful, e.g. move towards "happy" over 35 minutes. A recurring theme was that playlists tended to be repetitive. One solution would be to allow the jittering of playlist trajectories and to do this jittering in high-dimensional space. The low-dimensional path then specifies a prior in the high-dimensional music space which can be perturbed to explore alternative expressions of that path.

Post-evaluation:

An enhanced version with a text search function was distributed at the end of the study. The encouraging result was that a month later, 3 of 8 participants still returned to the interface on a regular basis – once every few days, with one user generating 68 new playlists in the following weeks.

Acknowledgments Partially supported by Danish Council for Strategic Research: CoSound project, 11-115328

7. REFERENCES

- [1] Dominikus Baur, Bernhard Hering, Sebastian Boring, and Andreas Butz. Who needs interaction anyway? Exploring mobile playlist creation from manual to automatic. *Proc. 16th Int. Conf. on Intelligent User Interfaces*, pages 291–294, 2011.
- [2] Dominikus Baur, Bartholomäus Steinmayr, and Andreas Butz. SongWords: Exploring music collections through lyrics. In *Proc. ISMIR*, pages 531–536, 2010.
- [3] Daniel Boland, Ross McLachlan, and Roderick Murray-Smith. Inferring Music Selections for Casual Music Interaction. *EuroHCIR*, pages 15–18, 2013.
- [4] Daniel Boland, Ross McLachlan, and Roderick Murray-Smith. Engaging with mobile music retrieval. In *MobileHCI 2015, Copenhagen*, 2015.
- [5] Daniel Boland and Roderick Murray-Smith. Information-theoretic measures of music listening behaviour. In *Proc. ISMIR, Taipei*, 2014.
- [6] Vincent Castaignet and Frederic Vavrilie. (23.04.2014) <http://musiccovery.com/>.
- [7] Martin Ester, Hans P Kriegel, Jorg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Second Int. Conf. on Knowledge Discovery and Data Mining*, pages 226–231, 1996.
- [8] Raphael A. Finkel and Jon Louis Bentley. Quad trees a data structure for retrieval on composite keys. *Acta Informatica*, 4(1):1–9, 1974.
- [9] Alinka Greasley, Alexandra Lamont, and John Sloboda. Exploring Musical Preferences: An In-Depth Qualitative Study of Adults' Liking for Music in Their Personal Collections. *Qualitative Research in Psychology*, 10:402–427, 2013.
- [10] Jiajun Zhu and Lie Lu. Perceptual Visualization of a Music Collection. In *2005 IEEE Int. Conf. on Multimedia and Expo*, pages 1058–1061. IEEE, 2005.
- [11] JungHyun Kim, Seungjae Lee, SungMin Kim, and Won Young Yoo. Music mood classification model based on arousal-valence values. *13th Int. Conf. on Advanced Comm. Technology*, pages 292–295, 2011.
- [12] Cyril Laurier, Mohamed Sordo, Joan Serrà, and Perfecto Herrera. Music mood representations from social tags. In *Proc. ISMIR*, pages 381–386, 2009.
- [13] Adam J Lonsdale and Adrian C North. Why do we listen to music? A uses and gratifications analysis. *British Journal of Psychology*, 102(1):108–134, 2011.
- [14] Matthias Lübbers, Dominik and Jarke. Adaptive Multimodal Exploration of Music Collections. In *Proc. ISMIR*, pages 195–200, 2009.
- [15] Gerald M. Murch. Physiological principles for the effective use of color. *Computer Graphics and Applications, IEEE*, 4(11):48–55, 1984.
- [16] Robert Neumayer, Michael Dittenbach, and Andreas Rauber. PlaySOM and PocketSOMPlayer, alternative interfaces to large music collections. In *Proc. ISMIR*, pages 618–623, 2005.
- [17] Elias Pampalk, Andreas Rauber, and Dieter Merkl. Content-based organization and visualization of music archives. *Proc. 10th ACM Int. Conf. on Multimedia*, page 570, 2002.
- [18] Henning Pohl and Roderick Murray-Smith. Focused and casual interactions: allowing users to vary their level of engagement. In *Proc. ACM SIGCHI Conf. on Human Factors in Computing Systems*, pages 2223–2232, 2013.
- [19] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006.
- [20] James A. Russell. A circumplex model of affect. *Journal of Personality and Social Psychology*, 39:1161–1178, 1980.
- [21] Markus Schedl and Arthur Flexer. Putting the User in the Center of Music Information Retrieval. In *Proc. ISMIR, Porto, Portugal*, 2012.
- [22] Devinderjit Sivia and John Skilling. Data Analysis: A Bayesian Tutorial. *Technometrics*, 40(2):155, 1998.
- [23] Sebastian Stober. *Adaptive Methods for User-Centered Organization of Music Collections*. PhD thesis, Otto-von-Guericke-Universität Magdeburg, 2011.
- [24] Bob L. Sturm. A simple method to determine if a music information retrieval system is a horse. *IEEE Transactions on Multimedia*, 16(6):1636–1644, 2014.
- [25] Laurens van der Maaten and Geoffrey Hinton. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.
- [26] Rob van Gulik and Fabio Vignoli. Visual playlist generation on the artist map. In *Proc. ISMIR*, pages 520–523, 2005.

CONCEPTUAL BLENDING IN MUSIC CADENCES: A FORMAL MODEL AND SUBJECTIVE EVALUATION.

Asterios Zacharakis
School of Music Studies,
Aristotle University of
Thessaloniki, Greece
aszachar@mus.auth.gr

Maximos Kaliakatsos-Papakostas
School of Music Studies,
Aristotle University of
Thessaloniki, Greece
maxk@mus.auth.gr

Emilios Cambouropoulos
School of Music Studies,
Aristotle University of
Thessaloniki, Greece
emilios@mus.auth.gr

ABSTRACT

Conceptual blending is a cognitive theory whereby elements from diverse, but structurally-related, mental spaces are ‘blended’ giving rise to new conceptual spaces. This study focuses on structural blending utilising an algorithmic formalisation for conceptual blending applied to harmonic concepts. More specifically, it investigates the ability of the system to produce meaningful blends between harmonic cadences, which arguably constitute the most fundamental harmonic concept. The system creates a variety of blends combining elements of the penultimate chords of two input cadences and it further estimates the expected relationships between the produced blends. Then, a preliminary subjective evaluation of the proposed blending system is presented. A pairwise dissimilarity listening test was conducted using original and blended cadences as stimuli. Subsequent multidimensional scaling analysis produced spatial configurations for both behavioural data and dissimilarity estimations by the algorithm. Comparison of the two configurations showed that the system is capable of making fair predictions of the perceived dissimilarities between the blended cadences. This implies that this conceptual blending approach is able to create perceptually meaningful blends based on self-evaluation of its outcome.

1. INTRODUCTION

Conceptual blending is a cognitive theory developed by Fauconnier and Turner [8] whereby elements from diverse, but structurally-related, mental spaces are ‘blended’ giving rise to new conceptual spaces that often possess new powerful interpretative properties allowing better understanding of known concepts or the emergence of novel concepts altogether. The general framework within which the current work is placed, comprises a formal model for conceptual blending [7] based on Goguen’s initial ideas of a Unified Concept Theory [9, 18]. This model incorporates im-

portant interdisciplinary research advances from cognitive science, artificial intelligence, formal methods and computational creativity. To substantiate its potential, a proof-of-concept autonomous computational creative system that performs melodic harmonisation is being developed.

Musical meaning is to a large extent self-referential; themes, motives, rhythmic patterns, harmonic progressions and so on emerge via self-reference rather than external reference to non-musical concepts. Since musical meaning largely relies on structure and since conceptual blending involves mapping between different conceptual structures, music seems to be an ideal domain for conceptual blending (musical cross-domain blending is discussed by Antović [1], Cook [6], Zbikowski [24]). Indeed, structural conceptual blending is omnipresent in music making: from individual pieces harmoniously combining music characteristics of different pieces/styles, to entire musical styles having emerged as a result of blending between diverse music idioms.

Suppose we have a basic tonal ontology where only diatonic notes are allowed and dissonances in chords are mostly forbidden (except possibly using minor 7th intervals as in the dominant seventh chord). We assume that some basic cadences have been established as salient harmonic functions around which the harmonic language of the idiom(s) has been developed, such as the authentic/perfect cadence, the half cadence, the plagal cadence and, even, older 15th century modal cadences such as the Phrygian cadence (Figure 1). The main question to be addressed is the following: Is it possible for a computational system to enrich its learned tonal ontology by inventing ‘new’ meaningful cadences based on blending between known cadences?

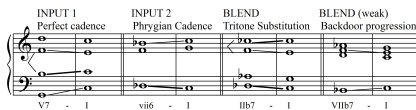


Figure 1: Conceptual blending between the basic perfect and Phrygian cadences gives rise to the Tritone Substitution progression/cadence. The Backdoor progression can also be derived as a weaker blend since less attributes of the two input spaces are retained leading to a lower rating by the system.

© Asterios Zacharakis, Maximos Kaliakatsos-Papakostas, Emilios Cambouropoulos.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Asterios Zacharakis, Maximos Kaliakatsos-Papakostas, Emilios Cambouropoulos, “Conceptual Blending in Music Cadences: A formal model and subjective evaluation.”, 16th International Society for Music Information Retrieval Conference, 2015.

Figure 1 presents a conceptual blending example where the perfect and the 15th century modal Phrygian cadences are used as input spaces. These have been chosen in this example as they are both final cadences to the tonic and at the same time, they are very different (i.e. the Phrygian mode does not have an upward leading note to the tonic but rather a downward ‘leading note’ from the IIb to the I). Initially, these cadences are formally described as simply pitch classes with reference to a tonal centre (C tonality was adopted in this case). Assuming that the final chord is always a common tonic chord, blending takes place by combining pitches of the penultimate chords between different cadences. Rather than mere combination of pitches, other characteristic attributes of a cadence are also taken into account. Weights/priorities that reflect relative prominence (e.g., root, upwards or downwards leading note, dissonant note that requires resolution – see Figure 1 where lines of variable thickness illustrate relative strength of voice-leading connections in cadences) are assigned to each chord note according to a human expert. The ‘blended’ penultimate chord is also constrained to comply with a certain chord type such as the major or minor chord (in this instance the characteristic major chord with minor seventh was preferred).

Let us examine the particular blending example between the perfect and the Phrygian cadences more closely. Notes from the two input penultimate chords of the two cadence types create a large number of possible combinations. We start with combinations (at least 3 notes) of the highest priority/salience notes (notes connected with bold lines in Figure 1). Many of these combinations are not triadic chords and may be filtered out using a set of constraints (in this instance our constraint is to have a chord that is a standard characteristic tonal chord such as a dominant seventh), while also a note completion step might be required (adding notes to incomplete blending results) if the examined combination incorporates too few notes; more details regarding constraints will be given in subsection 2.1. Among the accepted blends, the most highly rated one based on priorities values is the tritone substitution progression (IIb7-I) of jazz harmony. This simple blending mechanism ‘invents’ a chord progression that embodies characteristics of the Phrygian cadence (root/bass downward motion by semitone) and the dominant seventh chord (resolution of tritone). Thus, it creates a new harmonic ‘concept’ that was actually introduced in jazz, centuries later than the original input cadences. The Backdoor Progression also appears in the potential blends albeit with much lower priority (i.e. a much weaker blend) – see Figure 1. A number of other applications and uses of harmonic blending [11] and, more specifically, chord blending are reported in [7].

Following the above, a challenging question that needs to be addressed concerns the evaluation of the outcome produced by such a creative system. The mere definition of creativity is problematic and not commonly accepted as many authors approach it from different perspectives (e.g., [3,5,17,21], for a comprehensive discussion see [10]). Ap-

plications of computational creativity to music pose the extra issue of aesthetic quality judgement since creativity may not always be accompanied by aesthetic value and vice versa. In terms of assessing a creative system, the two usual approaches are to either directly evaluate the final product or to evaluate the productive mechanism [16]. The present work is concluded by an empirical experiment that attempts to address the former by shedding some light on how the system’s output is perceived leaving –for the moment– the issue of aesthetic value intact.

To this end, the output of the cadence blending system described in the following section is used to set up a preliminary subjective evaluation of the conceptual blending algorithm applied to cadence invention. As stated previously, the computational system is capable of creating a variety of blends combining elements of two input cadences and it further estimates the expected relationships between the produced blends. A number of blends between the perfect and the Phrygian cadence were produced in order to test the ability of the cadence blending system to accurately predict their perceived relations (i.e. the functionality of the blends) using an ‘objective’ distance metric (see subsection 2.2). To achieve this, a pairwise dissimilarity listening test for the nine cadences (two original, four blends and three miscellaneous) was designed and conducted. Subsequent multidimensional scaling (MDS) analysis was utilised to obtain geometric configurations for both behaviourally acquired pairwise distances and dissimilarity estimations by the algorithm. Comparison of the two configurations showed that the system can model the perceptual space quite accurately.

2. FORMAL CONCEPTUAL BLENDING MODEL

This section begins with a description of the conceptual blending mechanism utilised by the system for cadence construction. It then proceeds with a consideration of a naive distance metric for pairs of cadences based on representation of cadences according to the system.

2.1 Cadence generation through chord blending

A cadence is described as a progression of (at least) two chords that conclude a phrase, section or piece of music [2]. In our case we have examined the simplest case of two chords, a penultimate and a final chord. If the final – destination – chord is considered fixed, then blending between two cadences can occur by blending the penultimate chords of the cadences. The penultimate chords should therefore be described in a way that reflects the ‘functional’ role of their constitutive components. To this end, ‘chord-type’ properties of the penultimate chords (i.e. characteristics of type such as major, minor etc.) should be considered in combination with ‘key-related’ characteristics (i.e. their relations to the final chord). For instance, a ‘chord-type’ and distinctive characteristic of the penultimate chord in the perfect cadence (V7) is the fact that it includes a tritone (between the third and the minor seventh), while two ‘key-related’ important characteristics are a) the fact that

it includes the leading tone to the tonic (expressed as the pitch class 11 relative to the local key) and b) that its root moves by a perfect fifth to the tonic. Additionally, the specification of cadences (penultimate chords) should incorporate priority values, taking into account the fact that not all characteristics ('chord-type' or 'key-related') are equally salient.

The blending framework employed in this paper for producing novel cadences through concept blending has been presented in [7]. This framework follows Goguen's proposal to model conceptual spaces as algebraic specifications, while the utilised specifications defined in a variant of *Common Algebraic Specification Language* (CASL) [14] are extended with priority values associated to axioms. These specifications incorporate symbols as basic building blocks, over which more refined specifications are constructed, beginning from the sort 'Note' that is utilised to build the sort 'Chord'. The sort *Chord* represents the penultimate chord of the cadence which is in fact the notion of the cadence as previously described. A *Note* can receive values between 0 and 11, indicating the 12 pitch classes. In addition, a '+' operator is considered for arithmetics of addition in a modulo 12. For example, $7 + 9 = 4$ denotes that a sixth plus a fifth is a major third.

A *Chord* specification incorporates two kinds of attributes that relate to the aforementioned 'chord-type' and 'key-related' attributes, respectively '*chordNote*' and '*keyNote*'. The '*chordNote*' property indicates semitone distances between the chord's root and the notes comprising the chord, e.g., a major chord with minor seventh has the following relative notes: [0, 4, 7, 10]. On the other hand, the '*keyNote*' property indicates semitone distance between the scale's root note and the notes comprising the chord, e.g., a major chord with minor seventh and with chord root on the fifth degree of the major scale (i.e. pitch class 7) has the following key-related notes: [7, 11, 2, 5].

The salient characteristics of penultimate chords, and in extension of cadences, are defined for the two input spaces by employing human knowledge¹. The salience of a penultimate chord property is input to the system as a *priority* value which is then directly linked to this property. The output of conceptual blending, i.e. a conceptual blend, should incorporate the most salient features of the two input spaces – reflected by higher priority values. Additional constraints that concern further knowledge about chords are imposed. For the system employed in this paper, presented in more detail in [7], the additional constraints concern the facts that a chord should not have a major and a minor third ('*chordNote* 3 and 4) at the same time, it should not have a minor second ('*chordNote*' 1) and it should not have both a perfect and a diminished fifth ('*chordNote*' 6 and 7) at the same time. When a new *blendoid*² emerges, these constraints are enforced in the form of a *consistency*

¹ In this study, for convenience, they are determined manually by a music expert.

² The term *blendoid* refers to a possible result of blending, which, however, is not necessarily consistent or optimal. Additional criteria either validate or discard the consistency of a blendoid as well as evaluate it as optimal (based on 'blending optimality principles' or on domain-specific characteristics inherited to the blendoid).

check on the chord specification. Thereby, *inconsistent* blends are discarded.

The input cadences that have been selected to demonstrate blending of harmonic concepts were the *perfect* and the *Phrygian*, with their attributes and priorities depicted in Table 1. For both cadences, the highest priorities are assigned in such a way that the most musically salient aspects of the penultimate chords are highlighted. For the perfect cadence, the most highlighted features include the leading note (*keyNote*: 11) to the tonic and the fact that its type includes a tritone (*chordNote*: 4 and *chordNote*: 10). For the Phrygian cadence, the musically salient feature is the descending leading note (*keyNote*: 1) to the tonic.

perfect		Phrygian	
attribute	priority	attribute	priority
<i>keyNote</i> : 7	<i>p</i> : 2	<i>keyNote</i> : 10	<i>p</i> : 1
<i>keyNote</i> : 11	<i>p</i> : 3	<i>keyNote</i> : 1	<i>p</i> : 3
<i>keyNote</i> : 2	<i>p</i> : 1	<i>keyNote</i> : 5	<i>p</i> : 2
<i>keyNote</i> : 5	<i>p</i> : 2		
<i>chordNote</i> : 0	<i>p</i> : 1	<i>chordNote</i> : 0	<i>p</i> : 1
<i>chordNote</i> : 4	<i>p</i> : 3	<i>chordNote</i> : 3	<i>p</i> : 1
<i>chordNote</i> : 7	<i>p</i> : 1	<i>chordNote</i> : 7	<i>p</i> : 1
<i>chordNote</i> : 10	<i>p</i> : 3		

Table 1: Attributes and priorities (higher values indicate higher priority) considered in the blending system for the input penultimate chords in the perfect and Phrygian cadences. Common attributes of both cadences (the generic space [7]) appear in boxes.

tritone		backdoor	
attribute	priority	attribute	priority
<i>keyNote</i> : 1	<i>p</i> : 3	<i>keyNote</i> : 10	<i>p</i> : 1
<i>keyNote</i> : 11	<i>p</i> : 3	<i>keyNote</i> : 2	<i>p</i> : 1
<i>keyNote</i> : 5	<i>p</i> : 2	<i>keyNote</i> : 5	<i>p</i> : 2
<i>keyNote</i> : 8	<i>p</i> : 1	<i>keyNote</i> : 8	<i>p</i> : 1
<i>chordNote</i> : 0	<i>p</i> : 1	<i>chordNote</i> : 0	<i>p</i> : 1
<i>chordNote</i> : 4	<i>p</i> : 3	<i>chordNote</i> : 4	<i>p</i> : 3
<i>chordNote</i> : 7	<i>p</i> : 1	<i>chordNote</i> : 7	<i>p</i> : 1
<i>chordNote</i> : 10	<i>p</i> : 3	<i>chordNote</i> : 10	<i>p</i> : 3

Table 2: Attributes and priorities (higher values indicate higher priority) in the tritone substitution and backdoor cadences that result as blends from the perfect and Phrygian cadences. The completion step adds the *keyNote*: 8.

The computational chord blending framework combines the *salience* of chord features and core ideas of the notion of *Amalgams* [15], resulting in a process that iteratively produces blendoids with descending salience in their characteristics. However, the produced blendoids potentially require completion, i.e. additional reasoning mechanisms that fill-in incomplete properties. Let us consider the example of the tritone substitution cadence blend to elucidate the completion step, as demonstrated in Table 2. The tritone substitution cadence is acquired by preserving the most salient *keyNote* attributes (with priority 3) from both input spaces: [1, 5, 11], and all the *chordNote* attributes of the perfect cadence: [0, 4, 7, 10]. However,

the utilisation of the pitch classes [1, 5, 11] does not satisfy the requirements for a full dominant seventh chord of type [0, 4, 7, 10]. The completions step for the pilot study presented in [7] is performed manually, although it is possible to develop an automatic completion algorithm based on the chord root provided by the utilisation of the General Chord Type (GCT) [4] algorithm. For instance, in the tritone substitution, pitch class 1 is assigned as a root note, a fact that leads to the completion of the pitch class (*keyNote*): 8 as a perfect fifth (to match the *chordNote*: 7). The backdoor cadence preserves the *keyNote* attributes spaces: [2, 5, 10], which are not the ones with the highest priorities, and again all the *chordNote* attributes of the perfect cadence: [0, 4, 7, 10]. Similarly, the completion step assigns the pitch class 10 as a root note, while the requirement for a minor seventh (*chordNote*: 10) leads to importing the pitch class (*keyNote*): 8 into the blend. Since no background knowledge about the role of the attribute *keyNote*: 8 is given, the ‘default’ priority 1 is inserted, which will also be the case for all the examples in this paper: if attributes emerge through completion that have not been modelled in the input spaces, the default priority 1 is assigned.

2.2 Model-based distance metric

A naive method to compute the distances between pairs of cadences is by comparing their common features with the set of all their distinct features. In our case, since the final chord is always the same minor tonic, the comparison boils down to the features of their penultimate chords. Thereby, the more features these chords have in common, the more similar the cadences should be. For two cadences, C_i and C_j two sets are considered: the *intersection*, $\cap(C_i, C_j)$, and the *union*, $\cup(C_i, C_j)$ of their penultimate chord features. The intersection is the set of their common features and the union is the sum of all the features appearing in both cadences without repetitions. For instance, for the cadences indexed 1 and 3 in Table 3:

$$\cap(C_1, C_3) = [[5, 11], [0, 4, 7, 10]],$$

$$\cup(C_1, C_3) = [[1, 2, 5, 7, 8, 11], [0, 4, 7, 10]].$$

The considered distance based on the intersection and union of the features of penultimate chords is computed by dividing the number of elements in the intersection with the number of elements in the union. If $N(X)$ is the number of elements in a set X , then the distance between two cadences is computed as

$$d(C_i, C_j) = \frac{N(\cap(C_i, C_j))}{N(\cup(C_i, C_j))}.$$

In the aforementioned example, $d(C_1, C_3) = 6/10$.

3. EMPIRICAL EVALUATION

In order to investigate the functionality of the blended cadences (i.e. the perceived relationships between them) we

conducted a pairwise dissimilarity rating listening experiment using as stimuli the nine selected cadences described below. This approach is widely adopted in psychoacoustics because it enables the construction of perceptual spaces by employing multidimensional scaling (MDS) analysis on the obtained dissimilarity matrices.

3.1 Stimuli

The stimulus set consisted of the two input cadences (perfect and Phrygian), four blends of the input spaces and three miscellaneous cadences (Figure 2). More specifically, seven selected blends were as follows: blend 3 was the tritone substitution progression, blends 4 and 5 were the backdoor progression (the latter without seventh), cadence 6 was a plagal cadence (it was input manually as a cadence instance that was not a blend and was rather different to the two input cadences), cadence 7 contained a minor dominant penultimate chord, cadence 8 was essentially a French-sixth chord-type (similar in principle to the tritone substitution) and cadence 9 was a manually constructed non-blend chromatic chord. Note that all the cadences were assumed to be in C minor and each cadence was preceded by the notes C and F to reinforce perception of tonal context – the only chord that changed in each stimulus was the penultimate chord. Table 3 illustrates the cadences used in the subjective experiment with the *keyNote* and *chordNote* features grouped in two arrays. Therefore, since the system is able to produce blended cadences according to these features (*keyNote* and *chordNote*), the similarity between two cadences in terms of the system’s modelling should depend merely on them.

Figure 2: Score annotation of the two input cadences (1-2), 4 blends of the input spaces (3-6) and 3 miscellaneous cadences (7-9).

3.2 Participants

Fifteen listeners (aged 19-48, mean age: 26.5, 8 female) participated in the listening test. All reported normal hearing and long term music practice (years on average: 18.7, range: 6 to 43). Participants were students in the Department of Music Studies of the Aristotle University of Thessaloniki. All participants were naive about the purpose of the test.

index	input		blends				miscellaneous		
	1	2	3	4	5	6	7	8	9
<i>keyNote</i>	[7, 11, 2, 5]	[10, 1, 5]	[1, 5, 8, 11]	[10, 2, 5, 8]	[10, 2, 5]	[2, 5, 9, 0]	[7, 10, 2]	[1, 5, 7, 11]	[3, 7, 10, 1]
<i>chordNote</i>	[0, 4, 7, 10]	[0, 3, 7]	[0, 4, 7, 10]	[0, 4, 7, 10]	[0, 4, 7]	[0, 3, 7, 10]	[0, 3, 7]	[0, 4, 6, 10]	[0, 4, 7, 10]

Table 3: The penultimate cadence chords for the experiments along with their features and their respective indexes.

3.3 Procedure

In the pairwise dissimilarity listening test, participants were asked to compare all the pairs among the nine sound stimulus set using the free magnitude estimation method [23]. Therefore, they rated the perceptual distances of forty-five pairs (same pairs included) by freely typing in a number of their choice to represent dissimilarity of each pair (i.e., an unbounded scale) with 0 indicating a same pair.

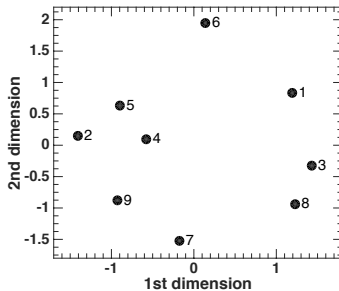
Listeners became familiar with the different cadences during an initial presentation of the stimulus set in random order. This was followed by a brief training stage where listeners rated four selected pairs of stimuli. For the main part of the experiment participants were allowed to listen to each pair of sounds as many times as needed prior to submitting their rating. The pairs were presented in random order and listeners were advised to retain a consistent rating strategy throughout the experiment. In total, the listening test sessions, including instructions and breaks, lasted around twenty minutes for most of the participants.

4. EXPERIMENTAL RESULTS

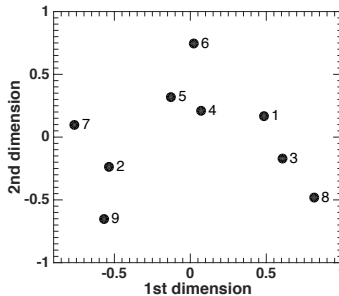
The proposed formal conceptual blending framework enables the generation of multiple cadences with different values of ‘importance’, as reflected by the priorities of the attributes preserved into the penultimate chords of the resulting cadences. For the purpose of this study, the system-wise ‘objective’ distance metric between cadences (see subsection 2.2) is merely based on the common features of the penultimate chords, not taking priority values into account. The aim of this study is to examine whether the pairwise distances between several cadences, as expressed by this ‘objective distance’ is aligned with the cognitive/perceptual distances that musically trained participants assign.

A non-metric, weighted individual differences scaling (INDSCAL) MDS analysis as offered by the SPSS PROXSCAL (proximity scaling) algorithm [13] was applied to the dissimilarity matrices. INDSCAL computes weights that represent the importance attributed to each perceptual dimension by each participant and then uses these weights to reconstruct a common perceptual space. Additionally, the ‘ordinal’ option applies a rank ordering transformation to the raw dissimilarities within each participant’s responses. The non-metric approach was adopted since it has been proven robust to the presence of monotonic transformations or random error in the data [19, 22].

A two-dimensional solution of the behavioural data with the following goodness of fit measures: Stress-I: .228



(a)



(b)

Figure 3: The perceptual (a) and the algorithmic (b) spatial configurations for the nine selected cadences. The cadences are labelled according to the indexes of table 3.

and Dispersion Accounted for (DAF): .947³ was favoured. Considering the number of objects in combination with the number of dimensions, the achieved Stress-I value does not imply an adequate fit between the MDS model produced disparities and the actual distances reported by the participants. This fact can be attributed to the high level of uncertainty present in the subjective responses. However, the satisfactory interpretability of the two dimensional configuration (as will be shown below) supports the acceptance of this solution.

The dissimilarity matrix that was produced by the distance metrics of the cadence-blending-system was also analysed through non-metric MDS. The two-dimensional

³ Stress-I is a measure of missfit where a lower value indicates a better fit (with a minimum of zero) and DAF is a measure of fit where a higher value indicates a better fit (with a maximum of one).

solution featured both acceptable Stress-I (.123) and DAF (.985). The configurations of both spaces are shown in Figure 3.

Visual inspection of the perceptual space reveals that prior expectations regarding cadence positioning are generally fulfilled. The perfect (no.1) and the Phrygian (no.2) input cadences are positioned far away from each other on the 1st dimension. This dimension could be interpreted as 'modal vs tonal' since negative values coincide with absence of the leading note [11] while positive values signify presence of the leading note. Cadences no.4 (backdoor with seventh) and 5 (backdoor without seventh) are naturally closely related. The clustering of no.4 and no.5 with the Phrygian could be explained by their shared notes [5, 10] and also by the absence of the leading note [11] that moves them away from the perfect cadence territory. Also, the close positioning of cadences no.3 (tritone substitution) and no.8 is explained by the fact that the former is a German-sixth-type while the latter is a French-sixth-type both sharing three basic notes [1, 5, 11]. These two cadences are additionally positioned more closely to the perfect cadence (no.1) than to the Phrygian showing that although the tritone substitution is created by incorporating the most salient attributes of the two input cadences (see subsection 2.1), it is not perceived as being equidistant between them. This can be explained by the fact that both no.3 and no.8 take the leading note [11] and the seventh [5] (that needs to be resolved) from the perfect cadence but only take note [1] (base of the Phrygian) from the Phrygian. Cadence no.6 -the plagal- is positioned in the middle between the perfect and Phrygian along dimension 1 but is expectedly an outlier along dimension 2.

The comparison between the perceptual and algorithmic configurations was performed using Tucker's congruence coefficient [20]. As a guideline, for the congruence coefficient, values larger than .92 are considered good/fair, and values larger than .95 practically show equality between configurations [12]. In our case, the congruence coefficient between the perceptual and the algorithmic space was computed to be .944 indicating that the system can make a very good estimation of the relationships between cadences.

5. CONCLUSIONS

According to the theory of conceptual blending developed by Fauconnier and Turner, novel conceptual spaces can be created by blending elements from diverse input conceptual spaces. Based on this theory and its category-theoretic interpretation proposed by Goguen this study presented initial developments of a system for blending between harmonic structures, using cadence blending as a proof-of-concept. To this end, two input spaces with simple formalisations of the perfect and the Phrygian cadences were used to produce several blended cadences.

The two input spaces along with the produced blends, and other cadences, were subjected to a pairwise dissimilarity rating listening test and subsequent MDS analysis in order to evaluate the output produced by the cadence

blending system. The basic aim of the study was to examine whether perceptual distances between pairs of cadences, as rated by the participants, were actually reflected by an objective distance metric that related to the formalisation of cadences in the blending system. Indeed, the comparative results showed that the system is capable of making fair predictions of the perceived dissimilarities between the blended cadences. Given the uncertainty introduced by both the demanding nature of the behavioural task and the MDS analyses for the two sets of data, this result is deemed rather satisfactory and leads to the following implications:

1. The presented cadence description framework is *meaningful*. Although the representation of knowledge in cadences is very elementary (just describing the penultimate chords with their absolute and relative notes), the derived results align with human perception/cognition.
2. The utilised blending methodology produces *consistent* results in the sense that resulting blends do indeed match the perceptual/cognitive attributes of the input spaces.

The utilisation of more sophisticated system-oriented metrics is expected to increase the accuracy of the self-evaluation process within the system so as to produce meaningful results for a wider combination of input cadences (also ending in different final chords) or even for more complex harmonic structures. As an obvious next step, the parameters of the system distance metric can be refined to optimise the fit between the algorithm's prediction and the actual perception of cadence dissimilarities.

Cadence blending is a proof-of-concept example of the computational framework for conceptual blending that is being developed in the context of the COINVENT project [18]. Overall, the results of the subjective experiment, even with this elementary representation of cadences, indicate the effectiveness of this framework towards creating meaningful output. The long term objective is the application of the computational blending approach for developing melodic harmonisation methodologies that facilitate structural blending between harmonies of diverse music idioms. This will require the development of ontologies capable of describing significantly more complex harmonic concepts compared to a simple harmonic cadence. At the same time, the employed subjective evaluation will need to be enriched by more elaborate experiments that will not only be able to assess the aesthetic value and functionality of the blends but to also address the challenge of rating longer stimuli.

6. ACKNOWLEDGEMENTS

This work is funded by the COINVENT project. The project COINVENT acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 611553.

7. REFERENCES

- [1] M. Antović. Musical metaphor revisited: Primitives, universals and conceptual blending. *Universals and Conceptual Blending*, 2011.
- [2] B. Benward and M. N. Saker. *Music in theory and practice*, volume I, page 359. McGraw-Hill, 7th edition, 2003.
- [3] M. A. Boden. *The creative mind: Myths and mechanisms*. Psychology Press, 2004.
- [4] E. Cambouropoulos, M. Kaliakatsos-Papakostas, and C. Tsougras. An idiom-independent representation of chords for computational music analysis and generation. In *Proceeding of the joint 11th Sound and Music Computing Conference (SMC) and 40th International Computer Music Conference (ICMC)*, ICMC-SMC 2014, 2014.
- [5] T. Collins, R. Laney, A. Willis, and P. H. Garthwaite. Developing and evaluating computational models of musical style. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing*, available on CJO2015:1–28, 2015.
- [6] N. Cook. Theorizing musical meaning. *Music Theory Spectrum*, 23(2):170–195, 2001.
- [7] M. Eppe, R. Confalonieri, E. Maclean, M. Kaliakatsos-Papakostas, E. Cambouropoulos, M. Schorlemmer, M. Codescu, and K.U. Kühnberger. Computational invention of cadences and chord progressions by conceptual chord-blending. In *International Joint Conference on Artificial Intelligence (IJCAI) 2015, accepted for publication*, 2015.
- [8] G. Fauconnier and M. Turner. *The Way We Think: Conceptual Blending And The Mind's Hidden Complexities*. Basic Books, New York, reprint edition, 2003.
- [9] J. Goguen. Mathematical Models of Cognitive Space and Time. In Daniel Andler, Yoshinori Ogawa, Mitsuhiro Okada, and Shigeru Watanabe, editors, *Reasoning and Cognition*, volume 2 of *Interdisciplinary Conference Series on Reasoning Studies*. Keio University Press, 2006.
- [10] A. K. Jordanous. *Evaluating computational creativity: a standardised procedure for evaluating creative systems and its application*. PhD thesis, University of Sussex, 2012.
- [11] M. Kaliakatsos-Papakostas, E. Cambouropoulos, K. Kühnberger, O. Kutz, and A. Smaill. Concept Invention and Music: Creating Novel Harmonies via Conceptual Blending. In *In Proceedings of the 9th Conference on Interdisciplinary Musicology (CIM2014)*, CIM2014, December 2014.
- [12] U. Lorenzo-Seva and J. M. F. Ten Berge. Tucker's congruence coefficient as a meaningful index of factor similarity. *Methodology: European Journal of Research Methods for the Behavioral and Social Sciences*, 2(2):57–64, 2006.
- [13] J. J. Meulman and W. J. Heiser. *PASW Categories 18, Chapter 3*. SPSS Inc., Chicago, 2008.
- [14] P. D. Mosses. *CASL Reference Manual – The Complete Documentation of the Common Algebraic Specification Language*, volume 2960. Springer, 2004.
- [15] S. Ontañón and E. Plaza. Amalgams: A Formal Approach for Combining Multiple Case Solutions. In *Proceedings of the 18th International Conference on Case-Based Reasoning Research and Development, ICCBR'10*, pages 257–271, Berlin, Heidelberg, 2010. Springer-Verlag.
- [16] M. Pearce and G. Wiggins. Towards a framework for the evaluation of machine compositions. In *Proceedings of the AISB'01 Symposium on Artificial Intelligence and Creativity in the Arts and Sciences*, pages 22–32, 2001.
- [17] M. Pearce and G. Wiggins. Evaluating cognitive models of musical composition. In *Proceedings of the 4th international joint workshop on computational creativity*, pages 73–80. Goldsmiths, University of London, 2007.
- [18] M. Schorlemmer, A. Smaill, K.U. Kühnberger, O. Kutz, S. Colton, E. Cambouropoulos, and A. Pease. Coinvent: Towards a computational concept invention theory. In *5th International Conference on Computational Creativity (ICCC) 2014*, June 2014.
- [19] R. N. Shepard. Metric structures in ordinal data. *Journal of Mathematical Psychology*, 3:287–315, 1966.
- [20] L. R. Tucker. A method for synthesis of factor analysis studies. Technical report, Washington, DC: Department of the Army., 1951.
- [21] G. A. Wiggins. A preliminary framework for description, analysis and comparison of creative systems. *Knowledge-Based Systems*, 19(7):449–458, 2006.
- [22] F. W. Young. Nonmetric multidimensional scaling: Recovery of metric information. *Psychometrika*, 35:455–473, 1970.
- [23] A. Zacharakis, K. Pasiadis, and J. D. Reiss. An interlanguage unification of musical timbre: bridging semantic, perceptual and acoustic dimensions. *Music Perception*, 32(4), 2015.
- [24] L. M. Zbikowski. *Conceptualizing Music: Cognitive Structure, Theory, and Analysis*. Oxford University Press, 2002.

HARMONIC-PERCUSSIVE SOURCE SEPARATION USING HARMONICITY AND SPARSITY CONSTRAINTS

Jeongsoo Park, Kyogu Lee

Music and Audio Research Group

Seoul National University, Seoul, Republic of Korea

{psprink, kglee}@snu.ac.kr

ABSTRACT

In this paper, we propose a novel approach to harmonic-percussive sound separation (HPSS) using Non-negative Matrix Factorization (NMF) with sparsity and harmonicity constraints. Conventional HPSS methods have focused on temporal continuity of harmonic components and spectral continuity of percussive components. However, it may not be appropriate to use them to separate time-varying harmonic signals such as vocals, vibratos, and glissandos, as they lack in temporal continuity. Based on the observation that the spectral distributions of harmonic and percussive signals differ – *i.e.*, harmonic components have harmonic and sparse structure while percussive components are broadband – we propose an algorithm that successfully separates the rapidly time-varying harmonic signals from the percussive ones by imposing different constraints on the two groups of spectral bases. Experiments with real recordings as well as synthesized sounds show that the proposed method outperforms the conventional methods.

1. INTRODUCTION

Recently, musical signal processing has received a great deal of attention especially with the rapid growth of digital music sales. Automatic musical feature extraction and analysis for a large amount of digital music data has been enabled with the support of computational power. The major purposes of such tasks include extracting musical information such as melody extraction, chord estimation, onset detection, and tempo estimation.

Because most music signals often consist of both harmonic and percussive signals, the extraction of tonal attributes is often severely degraded by the presence of percussive interference. On the other hand, when we analyze rhythmic attributes such as tempo estimation, the harmonic signals act as interference that may prevent accurate analysis. Consequently, the separation of harmonic and percussive components in music signals will function as an

important pre-processing step that allows efficient and precise analysis.

For these reasons, many researchers have focused on investigating HPSS using various approaches. Uhle *et al.* performed singular value decomposition (SVD) followed by independent component analysis (ICA) to separate drum sounds from the mixture [1]. Gillet *et al.* presented a drum-transcription algorithm based on band-wise decomposition using sub-band analysis [2].

Other researchers have employed matrix factorization techniques such as non-negative matrix factorization (NMF). Helen *et al.* proposed a two-stage process composed of a matrix-factorization step and a basis-classification step [3]. Kim *et al.* employed the matrix co-factorization technique, where spectrograms of the mixture sound and drum-only sound are jointly decomposed [4]. NMF with smoothness and sparseness constraints was utilized by Canadas-Quesada *et al.* [5]. The algorithm was developed based on assumptions regarding the anisotropic characteristics of the harmonic and percussive components; harmonic components have temporal continuity and spectral sparsity, whereas percussive components have spectral continuity and temporal sparsity.

Most HPSS algorithms have employed the same assumption. Ono *et al.* presented a simple technique to represent a mixture sound spectrogram as a sum of harmonic and percussive spectrograms based on the Euclidean distance [6]. Their technique aims to minimize the temporal dynamics of harmonic components and the spectral dynamics of percussive components. They further extended their work to use an alternative cost function based on the Kullback-Leibler (KL) divergence [7]. More recently, FitzGerald presented a median filtering-based algorithm [8], where a median filter is applied to the spectrogram in a row-wise and column-wise manner for the extraction of harmonic and percussive sounds, respectively. Gkiokas *et al.* also proposed a non-linear filter-based HPSS algorithm [9].

However, the assumption regarding the temporal continuity, which is considered to be crucial for conventional harmonic-percussive studies, does not account for the rapidly time-varying harmonic signals often present in vocal sounds and musical expressions such as slides, vibratos, or glissandos. This is because their spectrograms often fluctuate over short periods of time. Thus, it may degrade the performance of the algorithms, particularly when



© Jeongsoo Park, Kyogu Lee.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Jeongsoo Park, Kyogu Lee. "Harmonic-Percussive Source Separation Using Harmonicity and Sparsity Constraints", 16th International Society for Music Information Retrieval Conference, 2015.

loud vocal components or such musical expressions are mixed.

In this paper, we propose a HPSS algorithm that is classified as a spectrogram decomposition-based method. We consider the spectrum of harmonic components to have a harmonic and sparse structure in the frequency domain, whereas the spectrum of percussive components to have an unsparsely structure. To realize the successful separation of harmonic/percussive sounds, we apply constraints that impose a particular structure of the spectral bases. The novelty of the proposed method resides in the harmonic constraint, which is an extension of the sparsity constraint presented in previous works [10]. The constraint is closely related to the Dirichlet prior, which is frequently used in probabilistic analysis. Because the proposed algorithm does not assume temporal continuity for the separation of harmonic signals, we can successfully separate harmonic signals from the mixture sound, even when there are significant fluctuations over time.

The rest of this paper is organized as follows. Section 2 explains in detail how the proposed method works. In Section 3, we present experimental results, and in Section 4, we conclude the paper.

2. PROPOSED METHOD

In this section, we present a detailed explanation of the proposed HPSS method. The proposed algorithm uses the spectrogram-decomposition technique, NMF, with the harmonicity and sparsity constraints based on the Dirichlet prior. For the efficient description of the proposed method, we first introduce the conventional NMF. Then, the algorithm description for the proposed method is presented. Finally, the theoretical relations of the proposed method to the Dirichlet prior are described.

2.1 Conventional NMF

Lee and Seung introduced the multiplicative update rule of NMF for KL divergence [11]. As we iteratively update the parameters, we can represent a non-negative matrix, which may correspond to a magnitude spectrogram, as a multiplication of two non-negative matrices that may contain spectral bases and temporal bases. The update rule can be represented as:

$$\mathbf{H}_{k,n} \leftarrow \frac{\mathbf{H}_{k,n} \sum_m \{ \mathbf{W}_{m,k} \mathbf{F}_{m,n} / \tilde{\mathbf{F}}_{m,n} \}}{\sum_{m'} \mathbf{W}_{m',k}} \quad (1)$$

$$\mathbf{W}_{m,k} \leftarrow \frac{\mathbf{W}_{m,k} \sum_n \{ \mathbf{H}_{k,n} \mathbf{F}_{m,n} / \tilde{\mathbf{F}}_{m,n} \}}{\sum_{n'} \mathbf{H}_{k,n'}} \quad (2)$$

where \mathbf{F} and $\tilde{\mathbf{F}}$ denote the $M \times N$ magnitude spectrogram of an audio mixture, and its estimation, respectively, \mathbf{W} and \mathbf{H} denote the $M \times K$ matrix of the spectral bases and the $K \times N$ matrix of their activations.

2.2 Formulation of Harmonic-Percussive Separation

We present a modified NMF algorithm to impose the characteristics of harmonic/percussive sounds. The update rule is separately represented for the harmonic source basis and percussive source basis as follows:

$$\mathbf{H}_{k,n} \leftarrow \frac{\mathbf{H}_{k,n} \sum_m \{ \mathbf{W}_{m,k} \mathbf{F}_{m,n} / \tilde{\mathbf{F}}_{m,n} \}}{\sum_{m'} \mathbf{W}_{m',k}} \quad (3)$$

$$\mathbf{W}_{m,k} \leftarrow \frac{\mathbf{W}_{m,k} \sum_n \{ \mathbf{H}_{k,n} \mathbf{F}_{m,n} / \tilde{\mathbf{F}}_{m,n} \}}{\sum_{n'} \mathbf{H}_{k,n'}} \quad (4)$$

$$\mathbf{w}_k \leftarrow (1 - \gamma_H^H) \mathbf{w}_k + \gamma_H^H \text{fft}(\text{fft}(\mathbf{w}_k))^p, k \in \Phi_H \quad (5)$$

$$\mathbf{w}_k \leftarrow \max(\mathbf{w}_k, 0), k \in \Phi_H \quad (6)$$

$$\begin{cases} \mathbf{w}_k \leftarrow (1 - \gamma_S^H) \mathbf{w}_k + \gamma_S^H (\mathbf{w}_k)^q, k \in \Phi_H \\ \mathbf{w}_k \leftarrow (1 - \gamma_S^P) \mathbf{w}_k + \gamma_S^P (\mathbf{w}_k)^r, k \in \Phi_P \end{cases} \quad (7)$$

where Φ_H and Φ_P denote a set of harmonic bases and percussive bases, respectively, $\text{fft}(\cdot)$ and $\text{ifft}(\cdot)$ denote the functions of the fast Fourier transform (FFT) and the inverse FFT (IFFT), respectively, \mathbf{w}_k denotes the k th column of \mathbf{W} , γ_H^H denotes the harmonicity weight parameter for the harmonic signal, and γ_S^H and γ_S^P denote the sparsity weight parameters for harmonic and percussive signals, respectively. Note that Eqns (3) and (4) are identical to Eqns (1) and (2), respectively. Eqns (5)-(7) contribute to shaping the spectral bases as desired as the iteration proceeds.

Mixing weights that have values between 0 and 1 represent the importance of each constraint imposition, and indicate the degree to which we need to impose the characteristic. To enable the harmonic bases to have a harmonic and sparse structure while preserving the original figures of spectral bases, γ_H^H and γ_S^H are set to have small positive numbers, as the effect of the constraint is accumulated over the iteration.

The exponents p , q , and r have to be determined considering the range of each parameter, $0 \leq r \leq 1 \leq p, q$. Here, p and q respectively reflect the degree of harmonicity and sparsity of the destination, and they have to be controlled considering the spectral characteristics of the original harmonic sources. Likewise, r reflects the degree of “unsparsity” of the percussive sources.

Among the update equations shown above, the function of the conventional NMF update equations in Eqns (3) and (4) is to minimize the error between \mathbf{F} and its estimation $\tilde{\mathbf{F}}$. On the other hand, the remainders of the equations aim to shape the spectral bases. The sparsity constraint in Eqn (7) has been similarly adopted for the matrix decomposition [10], and it is based on the fact that the square operation increases the differences among the vector components. If the square root operation is used instead, as

in the percussive case of Eqn (7), unsparsity can be imposed to the basis. Similarly, we can extend this concept to the harmonicity. The second term in Eqn (5) denotes the harmonics-emphasized basis, which is due to the fact that the *spectrum of the spectrum* is sparse. To prevent elements from being negative, the $\max(\cdot, \cdot)$ operation in Eqn (6) has to be jointly involved.

The harmonic and percussive sounds are reconstructed using the corresponding bases as follows:

$$\mathbf{F}^{(Harmonic)} = \sum_{k \in \Phi_H} \mathbf{w}_k \mathbf{h}_k \quad (8)$$

$$\mathbf{F}^{(Percussive)} = \sum_{k \in \Phi_P} \mathbf{w}_k \mathbf{h}_k \quad (9)$$

where \mathbf{h}_k denotes the k th row of \mathbf{H} .

2.3 Relation to Dirichlet Prior

The proposed update equations can be intuitively comprehended. However, the equations are based on a firm theoretical background, not heuristically induced. In this subsection, we employ Dirichlet prior from the probability theory, and investigate its relations to the proposed method.

Priors were primarily adopted for the Bayesian probability theory, including the probabilistic latent component analysis (PLCA) or probabilistic latent semantic analysis (PLSA). Such spectrogram decomposition techniques often regard spectrogram components as histogram elements of multinomial distributions. Because the Dirichlet distribution is a conjugate prior of a multinomial distribution, it can be adopted as a prior knowledge of a multinomial distribution. By adopting the prior, we can modify our goal to be the maximizing posterior from the maximizing likelihood. For this reason, the Dirichlet prior has been adopted for the matrix factorization in the previous works [10], [12]. Our method employs one of the extensions of the Dirichlet prior for harmonicity imposition.

Because PLCA is a special case of NMF, where its cost function is KL divergence [13], we can generalize the Dirichlet prior of the PLCA [12] by applying it to the NMF algorithm as follows:

$$\mathbf{H}_{k,n} \leftarrow (1 - \gamma_1) \frac{\mathbf{H}_{k,n} \sum_m \left\{ \mathbf{W}_{m,k} \mathbf{F}_{m,n} / \tilde{\mathbf{F}}_{m,n} \right\}}{\sum_{m'} \mathbf{W}_{m',k}} + \gamma_1 \mathbf{A}_{k,n} \quad (10)$$

$$\mathbf{W}_{m,k} \leftarrow (1 - \gamma_2) \frac{\mathbf{W}_{m,k} \sum_n \left\{ \mathbf{H}_{k,n} \mathbf{F}_{m,n} / \tilde{\mathbf{F}}_{m,n} \right\}}{\sum_{n'} \mathbf{H}_{k,n'}} + \gamma_2 \mathbf{B}_{m,k} \quad (11)$$

where \mathbf{A} and \mathbf{B} denote the matrices of hyper parameters with respect to \mathbf{H} and \mathbf{W} , respectively, and γ_1 and γ_2 denote the mixing weights. In our research, we focus only on the spectral bases, and thus Eqn (10) is discarded. As can be observed, the proposed update equations, Eqns (3)-(7),

have the same form as Eqn (11), and the way in which we shape the spectral bases depends on the form of \mathbf{B} matrix.

Frequency-domain sparsity imposition can be easily achieved by setting the hyper parameter \mathbf{B} as [10]

$$\mathbf{b}_k = (\mathbf{w}_k)^u \quad (12)$$

where \mathbf{b}_k denotes the k th column of \mathbf{B} , and u denotes an exponent that controls the degree of sparsity of \mathbf{b}_k .

On the other hand, harmonicity imposition can be achieved when the hyper parameter is represented as

$$\mathbf{b}_k = \text{ifft}(\{\text{fft}(\mathbf{w}_k)\}^v) \quad (13)$$

where v denotes the exponent that controls the degree of harmonicity of \mathbf{b}_k . This is because a periodic signal can be represented as a sum of sinusoids, and the spectrum of the periodic signal is sparse. Conversely, if a spectrum is sparse, we can assume that the original signal has a strongly periodic characteristic. Thus, we aim to make the *spectrum of the spectrum* to be sparse in order to shape a signal such that it has a harmonic structure. Note that in order to prevent destructive interference caused by phase distortion, we have to manipulate only the magnitudes within the IFFT function, preserving the original phases of $\text{fft}(\mathbf{w}_k)$.

3. PERFORMANCE EVALUATION

3.1 Sample Problem

In this section, we apply the proposed method and the conventional methods to simple sample examples, which is suitable for showing the novelty and validity of the proposed method. Spectrograms of synthesized sounds that consist of horizontal and vertical lines are presented in Figure 1(a) and Figure 2(a). Figure 1(a) models the case where a pitched harmonic sound is sustained for a certain period. The sounds of harmonic instruments such as guitars, pianos, flutes, and violins fall within this scenario. On the other hand, Figure 2(a) illustrates the case where a harmonic signal alters its frequency over time. In this case, vibratos, glissandos, and vocal signals correspond to the harmonic components. We compare the performance of the proposed method to the separation results obtained using three conventional methods: Ono *et al.*'s Euclidean distance-based method [6], Ono *et al.*'s KL divergence-based method [7], and FitzGerald's method [8].

As shown in Figure 1(b), both the conventional methods and the proposed method are able to successfully separate the sounds. This is because the horizontal lines in this example have horizontally continuous characteristics, which are assumed by the conventional methods to be present. However, when the harmonic sound vibrates and the horizontal lines fluctuate, as shown in Figure 2(a), conventional methods cannot distinguish the horizontal lines from vertical lines. As we can see in Figure 2(b), the estimated percussive components of conventional methods contain harmonic partials, and only the proposed method can successfully separate them. Thus, we can claim that the pro-

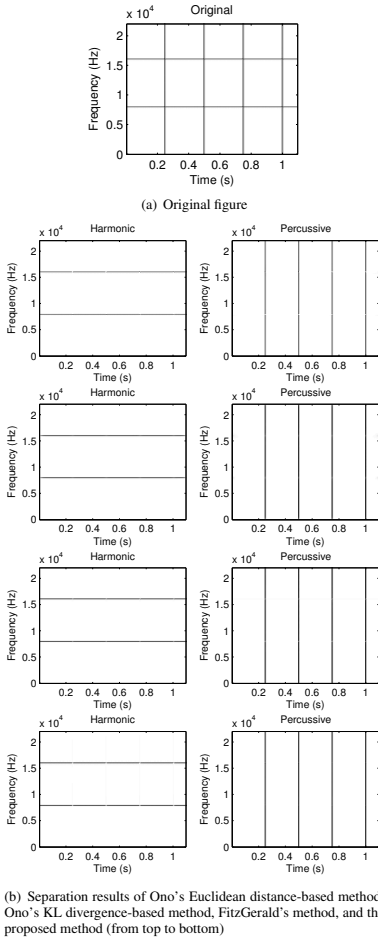


Figure 1. Sample example of separating horizontal lines and vertical lines.

posed method is not affected by variations in the pitch because it relies on the harmonic structure of the vertical axis, and not the degree of horizontal transition.

3.2 Qualitative Analysis

We evaluated the performance of the proposed method using a real recording example. Figure 3 shows a log-scale plot of the spectrogram of an excerpt from “Billie Jean,” by Michael Jackson. The signal was sampled at 22,050 Hz, and the frame size and overlap size were set to 1,024 and 512, respectively. We can observe from the spectrogram

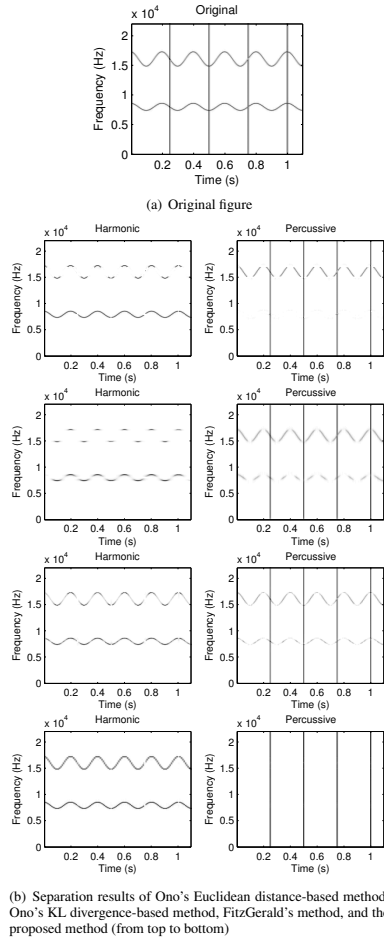


Figure 2. Sample example of separating fluctuating horizontal lines and vertical lines.

that the excerpt contains both harmonic and percussive components. The harmonic components can be seen as horizontally connected lines, whereas the percussive components are seen as vertical lines as in the sample examples.

Figure 4(a) and (b) show the separation results of the harmonic sound (up) and percussive sound (down), which were obtained using Ono *et al.*'s Euclidean distance-based method and KL divergence-based method, respectively. Here, we set the parameters to the values recommended in the references. We observe that the estimated percussive

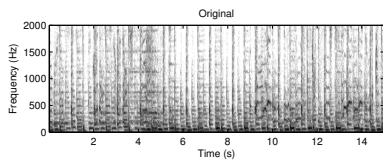


Figure 3. Spectrogram of a real audio recording example (“Billie Jean” by Michael Jackson).

components still contain harmonic components that may correspond to the vocal components. This is because Ono *et al.*'s algorithms aim to minimize the temporal transition of the harmonic spectrogram. However, vocal components in the original spectrogram do not match well with the underlying assumption.

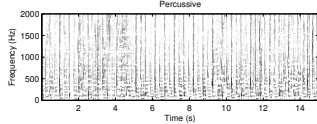
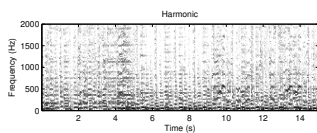
Figure 4(c) shows the result of FitzGerald's method with a median filter length of 17 and when the exponent for the Wiener filter-based soft mask is two, as recommended by FitzGerald [8]. We also observe that the separated percussive components still contain harmonic components, as in the previous case. This is because of the use of a one-dimensional median filter, which assumes that the harmonic components are sustained for several periods.

Figure 4(d) shows the performance of the proposed method. We observe that the harmonic and percussive components are clearly separated, and the percussive components do not have any vocal components in these results. This is because unlike conventional methods, the proposed algorithm does not rely on the horizontal continuity principle. Rather, the proposed algorithm tries to account for the harmonic components using the harmonic and sparse spectral bases.

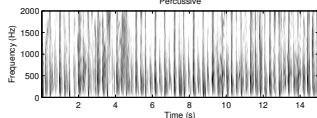
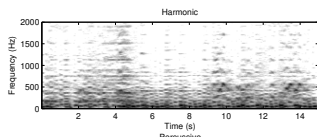
3.3 Quantitative Analysis

We performed a quantitative analysis to verify the validity of the proposed algorithm. First, we compiled a dataset that consists of 10 audio samples, which is a subset of the MASS database [14], but two sets of data, namely *tamy-que-pena-tanto-faz_6-19* and *tamy-que-pena-tanto-faz_46-57*, were excluded in this experiment because they lack percussive signals. Then, we obtained a spectrogram for each audio sample with the frame size and hop size set to 2,048 samples and 1,024 samples, respectively. Note that the sampling rate of the songs in the MASS dataset is 44,100 Hz. Finally, we measured the signal-to-distortion ratio (SDR), signal-to-interference ratio (SIR), and signal-to-artifact ratio (SAR) using the BSS_EVAL toolbox (http://bass-db.gforge.inria.fr/bss_eval/) supported by [15]. Table 1 shows the parameter values of the proposed method used in this experiment. The parameters of the conventional methods are set to the recommended values, as in the previous experiment.

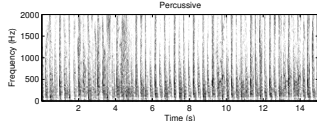
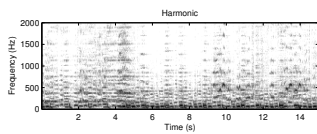
The evaluation results are summarized in Figure 5. We can see that the proposed method guarantees a better av-



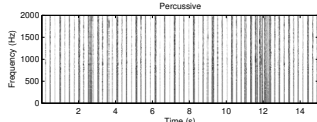
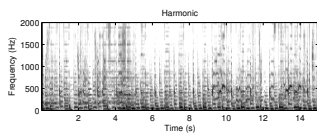
(a) Ono's Euclidean distance-based method



(b) Ono's KL divergence-based method



(c) FitzGerald's method



(d) Proposed method

Figure 4. Qualitative performance comparison of conventional and proposed methods.

verage SDR result compared to conventional methods, even though the proposed method has a lower SIR performance than Ono *et al.*'s Euclidean distance-based method. This is because the proposed method far outperforms other methods with respect to the SAR, which has a trade-off relation with the SIR [16].

Parameter	Value
p	1.1
q	1.1
r	0.5
γ_H^H	0.001
γ_S^H	0.001
γ_S^r	0.1
Number of bases (H,P)	(300,200)

Table 1. Experimental parameters.

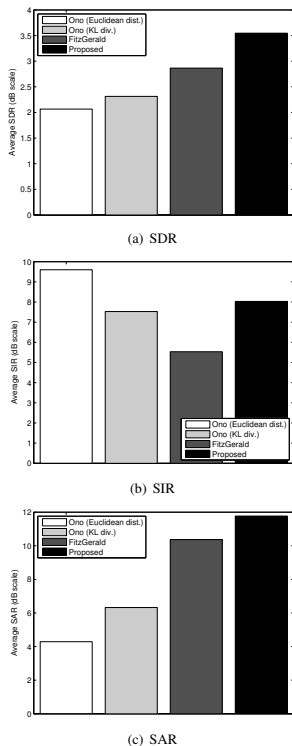


Figure 5. Quantitative performance comparison of conventional and proposed methods.

4. CONCLUSION

In this paper, we proposed a novel HPSS algorithm based on NMF with harmonicity and sparsity constraints. Conventional methods assumed that the harmonic components were represented as horizontal lines with temporal continuity. However, such an assumption could not be applied to the vocal components or various musical expressions of harmonic instruments. To overcome this problem, we presented a harmonicity constraint, which is a generalized Dirichlet prior. By letting the spectrum of the spectrum be harmonic and sparse, we could refine the harmonic components and eliminate inharmonic components. The experimental results showed the validity of the proposed method by comparing it with conventional methods.

5. ACKNOWLEDGEMENTS

This research was partly supported by the MSIP (Ministry of Science, ICT and Future Planning), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2015-H8501-15-1016) supervised by the IITP (Institute for Information & communications Technology Promotion). Also, this research was supported in part by the A3 Foresight Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology.

6. REFERENCES

- [1] C. Uhle, C. Dittmar, and T. Sporer, "Extraction of drum tracks from polyphonic music using independent subspace analysis," *Proceedings of the ICA*, pp. 843–847, April, 2003.
- [2] O. Gillet and G. Richard, "Drum track transcription of polyphonic music using noise subspace projection," *Proceedings of the ISMIR*, pp. 92–99, September, 2005.
- [3] M. Helen and T. Virtanen, "Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine," *Proceedings of the EUSIPCO*, September 2005.
- [4] M. Kim, J. Yoo, K. Kang, and S. Choi, "Nonnegative matrix partial co-factorization for spectral and temporal drum source separation," *IEEE Journal of Selected Topics in Signal Processing*, Vol.5, No.6, pp. 1192–1204, 2011.
- [5] F. J. Canadas-Quesada, P. Vera-Candeas, N. Ruiz-Reyes, J. Carabias-Orti, and P. Cabanas-Molero, "Percussive/harmonic sound separation by non-negative matrix factorization with smoothness/sparseness constraints," *EURASIP Journal on Audio, Speech, and Music Processing*, Vol.2014, No.1, pp. 1–17, 2014.
- [6] N. Ono, K. Miyamoto, J. Le Roux, H. Kameoka, and S. Sagayama, "Separation of a monaural audio signal

- into harmonic/percussive components by complementary diffusion on spectrogram,” *Proceedings of the EUSIPCO*, August, 2008.
- [7] N. Ono, K. Miyamoto, H. Kameoka, and S. Sagayama, “A real-time equalizer of harmonic and percussive components in music signals,” *Proceedings of the ISMIR*, pp. 139–144, September, 2008.
- [8] D. FitzGerald, “Harmonic/percussive separation using median filtering,” *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx-10)*, 2010.
- [9] A. Gkiokas, V. Papavassiliou, V. Katsouros, and G. Carayannis, “Deploying nonlinear image filters to spectrogram for harmonic/percussive separation,” *Proceedings of the 15th International Conference on Digital Audio Effects (DAFx-12)*, 2012.
- [10] M. Kim and P. Smaragdis, “Manifold preserving hierarchical topic models for quantization and approximation,” *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 1373–1381, 2013.
- [11] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” *Proceedings of the Advances in Neural Information Processing Systems*, November, 2000.
- [12] P. Smaragdis and G. J. Mysore, “Separation by humming: User-guided sound extraction from monophonic mixtures,” *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, October, 2009.
- [13] C. Ding, T. Li, and W. Peng, “On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing,” *Computational Statistics & Data Analysis*, Vol.52, No.8, pp. 3913–3927, 2008.
- [14] M. Vinyes, “MTG MASS database,” <http://www.mtg.upf.edu/static/mass/resources>, 2008.
- [15] E. Vincent, R. Gribonval, and C. Fevotte, “Performance measurement in blind audio source separation,” *IEEE Transactions on Audio, Speech, and Language Processing*, Vol.14, No.4, pp. 1462–1469, 2006.
- [16] D. L. Sun, and G. J. Mysore, “Universal speech models for speaker independent single channel source separation,” *Proceedings of the ICASSP*, 2013.

A HIERARCHICAL BAYESIAN FRAMEWORK FOR SCORE-INFORMED SOURCE SEPARATION OF PIANO MUSIC SIGNALS

Wai Man SZETO

Office of University General Education
The Chinese University of Hong Kong
wmszeto@cuhk.edu.hk

Kin Hong WONG

Department of Computer Science and Engineering
The Chinese University of Hong Kong
kh Wong@cse.cuhk.edu.hk

ABSTRACT

Here we propose a score-informed monaural source separation system to extract every tone from a mixture of piano tone signals. Two sinusoidal models in our earlier work are employed in the above-mentioned system to represent piano tones: the General Model and the Piano Model. The General Model, a variant of sinusoidal modeling, can represent a single tone with high modeling quality, yet it fails to separate mixtures of tones due to the overlapping partials. The Piano Model, on the other hand, is an instrument-specific model tailored for piano. Its modeling quality is lower but it can learn from training data (consisting entirely of isolated tones), resolve the overlapping partials and thus separate the mixtures. We formulate a new hierarchical Bayesian framework to run both Models in the source separation process so that the mixtures with overlapping partials can be separated with high quality. The results show that our proposed system gives robust and accurate separation of piano tone signal mixtures (including octaves) while achieving significantly better quality than those reported in related work done previously.

1. INTRODUCTION

Here we propose a score-informed monaural source separation system under a new hierarchical Bayesian framework to extract every tone from a mixture of piano tone signals with high separation quality. Two sinusoidal models in our earlier work in [14, 15] are employed in the above mentioned system to represent piano tones. Sinusoidal modeling is commonly used in many existing monaural source separation systems to model pitched musical sounds [6, 7, 9, 11, 16]. The major difficulty of source separation (SS) is to resolve overlapping partials.

Existing systems are based on assumptions on the general properties of pitched musical sounds. For example, the spectral envelope of tones is assumed to be smooth (as in [7, 16]), or that the amplitude envelope of each partial from the same note tends to be similar [11] (known as common amplitude modulation (CAM)), or that the amplitude envelope of a partial evolves similarly among different notes of the same musical instrument

in [9]. Yet these assumptions may not be suitable for SS of piano mixtures as explained in [15]. A very recent work in [17] can resolve two closed partials but it may not work on octaves, in which the partials of the upper tone are totally immersed within the frequencies of the lower tone. Moreover, it assumes that partials are exact multiples of the fundamental frequency. This assumption is not valid for piano because piano tones are only quasi-harmonic [1].

Instead of formulating similar assumptions, we limit input mixtures to piano music signals. This allows us to design a piano-specific model called the Piano Model (PM) to resolve overlapping partials in [15]. In piano music, a particular pitch tends to appear more than once. The tones of the same pitch share some common characteristics which can be captured by PM. Our system is based on two requirements. First, the pitches in the mixtures should reappear as isolated tones in the target recording. Second, the piano music is performed without pedaling. Then the isolated tones can be used as the training data for PM to resolve the overlapping partials even for octaves.

Although PM can resolve the overlapping partials, its modeling quality of single piano tones is lower than our General Model (GM) in [14]. However, GM cannot be directly applied to SS because it fails to separate mixtures of tones due to the overlapping partials. Here we formulate a new hierarchical Bayesian framework to run both PM and GM in the SS process so that the mixtures with overlapping partials can be separated with high quality. The separation process is divided into the training stage and the SS stage. Given the estimated PM parameters and the training data, we can, in the SS stage, set the prior distributions of the GM parameters to favor the proper regions of values under the Bayesian framework, estimate the GM parameters successfully even the case of overlapping partials, and reconstruct the individual tones in the mixtures with high quality. We hope that our system could shed some light on the empirical study of expressiveness in music performance [5] by comparing the subtleties of various artists' performances, based on individual tones extracted by SS.

2. SIGNAL MODELS

Here an individual tone (the sound of hitting one piano key) is considered as a particular sound source of the corresponding pitch. When multiple piano keys are pressed, a mixture signal is generated. We model a mixture signal as a sum of its corresponding individual tones as $y(t) = \sum_{k=1}^K x_k(t)$ where $y(t)$ is the observed mixture signal in the time domain, K is the



© Wai Man SZETO, Kin Hong WONG.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Wai Man SZETO, Kin Hong WONG. "A Hierarchical Bayesian Framework for Score-Informed Source Separation of Piano Music Signals", 16th International Society for Music Information Retrieval Conference, 2015.

number of tones in the mixture, $x_k(t)$ is the k th individual tone in the mixture, and t is the time in seconds. We assume that the score has been known so that the pitch and the duration of each $x_k(t)$ are given (music transcription systems [2, 10] can be used here). The goal of our research is to recover the signal of each individual tone $x_k(t)$ from the mixture signal $y(t)$ via the signal models GM and PM.

2.1 General Model (GM)

In [14], we present a frame-wise sinusoidal model called GM to represent a piano tone. For a piano tone, the frequencies of the partials are stable so the frequencies can be fixed across frames. The number of partials can also be fixed for a tone. In GM, the estimated tone $\hat{x}_{k,r}$, which is the estimate of the k th tone in a mixture at the r th frame, can be written as:

$$\hat{x}_{k,r}[l] = \sum_{m=1}^{M_k} w[l] (\alpha_{k,m,r} \cos(2\pi f_{k,m} t_l) + \beta_{k,m,r} \sin(2\pi f_{k,m} t_l)) \quad (1)$$

where M_k is the number of partials, $\alpha_{k,m,r}$ is the amplitude of the cosine component, $\beta_{k,m,r}$ is the amplitude of the sine component, $f_{k,m}$ is the frequency, $w[l]$ is the window function with the window length L and $l=0, \dots, L-1$, and t_l is the time in second at the index l so $t_l = l/f_s$ and f_s is the sampling frequency in Hz. The overlap-and-add method in [18] can be used to reconstruct the entire signal from GM.

Based on the above model, the estimated mixture $\hat{y}_r[l]$ at the r th frame is the sum of each estimated tone $\hat{x}_{k,r}[l]$ such that $\hat{y}_r[l] = \sum_{k=1}^K \hat{x}_{k,r}[l]$. The observed mixture is the sum of the estimated mixture and the noise term so $y_r[l] = \hat{y}_r[l] + v_r[l]$ where $v_r[l]$ is the noise component. To estimate the parameters in each frame, it is convenient to rewrite the model in (1) into the matrix form. Let \mathbf{H}_k be the frequency matrix of the k th tone in the form of

$$H_k[l, u] = \begin{cases} w[l] \cos(2\pi f_{k,u} t_l) & \text{if } 1 \leq u \leq M_k, \\ w[l] \sin(2\pi f_{k,u-M_k} t_l) & \text{if } M_k+1 \leq u \leq 2M_k \end{cases} \quad (2)$$

and we also let \mathbf{f}_k be the frequency vector containing all $f_{k,u}$.

The amplitudes of the cosine and sine terms of the k th tone at the r th frame can be expressed as a column vector $\mathbf{g}_{k,r}$ defined by

$$g_{k,r}[u] = \begin{cases} \alpha_{k,u,r} & \text{if } 1 \leq u \leq M_k, \\ \beta_{k,u-M_k,r} & \text{if } M_k+1 \leq u \leq 2M_k \end{cases}. \quad (3)$$

For the mixture, the frequency matrices from each tone are concatenated into the matrix $\mathbf{H} = [\mathbf{H}_1 \dots \mathbf{H}_K]$ and all \mathbf{f}_k are concatenated into the column vector $\mathbf{f} = [\mathbf{f}_1^T \dots \mathbf{f}_K^T]^T$. The amplitude vectors of each tone can also be concatenated into a column vector $\mathbf{g}_r = [\mathbf{g}_{1,r}^T \dots \mathbf{g}_{K,r}^T]^T$. The estimated mixture at r th frame can be expressed as $\hat{\mathbf{y}}_r = \mathbf{H} \mathbf{g}_r$, and the estimated mixture is related to the observed mixture as below:

$$\mathbf{y}_r = \mathbf{H} \mathbf{g}_r + \mathbf{v}_r \quad (4)$$

where \mathbf{v}_r is the noise term. It is modeled as the zero-mean Gaussian noise with the variance σ_v^2 .

The observed mixture signal can be expressed in the form of $\mathbf{Y} = [\mathbf{y}_1 \dots \mathbf{y}_R]$. Then the estimated mixture for all frames

can be written as

$$\hat{\mathbf{Y}} = \mathbf{H} \mathbf{G} \quad (5)$$

where $\hat{\mathbf{Y}} = [\hat{\mathbf{y}}_1 \dots \hat{\mathbf{y}}_R]$, $\mathbf{G} = [\mathbf{g}_1 \dots \mathbf{g}_R]$ and R is the number of frames. All GM parameters can be grouped into $\Theta = \{\mathbf{f}, \mathbf{G}\}$. The goal of our SS is to estimate both the frequency matrix \mathbf{H} and the amplitude matrix \mathbf{G} so that each individual tone can be reconstructed. However, \mathbf{H} is often rank deficient. This happens when some of the partials from different tones in the mixture are overlapping. This implies that if only the mixture in such case is given, it is impossible to separate the mixture into its individual tones unless more information is provided. This problem can be solved by using the training data as the prior information under the Bayesian framework in Section 3.

2.2 Piano Model (PM)

In [15], we propose PM to resolve the overlapping partials by exploring the common properties of recurring tones. PM employs a time-varying sum-of-sinusoidal signal model for piano tones, and it describes a tone in an entire duration instead of a single analysis frame as

$$\hat{x}_k(t_n) = \sum_{m=1}^{M_k} a(t_n; c_k, \varphi_{k,m}) \cdot \cos(2\pi f_{k,m} t_n + \phi_{k,m}) \quad (6)$$

where M_k is the number of partials of the k th tone, $f_{k,m}$ and $\phi_{k,m}$ are the frequency and the phase respectively, and $a(t_n; c_k, \varphi_{k,m})$ is the time-varying amplitude of the partial stated in [15] where the envelope parameters $\varphi_{k,m}$ control the envelope surface against the intensity c_k and the time t_n . The intensity c_k is assigned to be the peak amplitude of the observed time-domain signal of the tone. The onset of each tone in the mixture may not be exactly the same so a time-shift factor is introduced for each tone in the estimated mixture $\hat{y}(t_n) = \sum_{k=1}^{M_k} \hat{x}_k(t_n - \tau_k)$ where τ_k is the time shift in seconds.

All parameters in PM for the k th tone can be grouped into a parameter set ψ_k so $\psi_k = \{\varphi_{k,m}, f_{k,m}, \phi_{k,m}, c_k, \tau_k\}$ and $\Psi = \{\psi_1, \dots, \psi_K\}$. The PM parameters ψ_k can be divided into two groups: the invariant PM parameters $\psi_{k,\bar{v}} = \{\varphi_{k,m}, f_{k,m}, \phi_{k,m}\}$ and the varying PM parameters $\psi_{k,v} = \{c_k, \tau_k\}$. The invariant PM parameters contain parameters invariant to instances of the same pitch and they are estimated from the training data. The varying PM parameters consist of parameters which may vary across instances. Given a mixture, only the varying PM parameters of the mixture are required to be estimated if the invariant PM parameters have been estimated from the training data.

In both GM and PM, we have assumed that the number of partials M_k of each tone is known. The number of partials M_k is fixed for all experiments. The details of finding M_k can be found in [14].

3. BAYESIAN FRAMEWORK FOR SS

This section will explain how the Bayesian framework integrates the two models in the previous section and incorporates the training data to resolve overlapping partials. Given the mixture \mathbf{y} and the training data \mathcal{X} , the goal of Bayesian SS with GM is to find the *Maximum A Posterior* (MAP) solution $\hat{\Theta}_y$ that maximizes the posterior $p(\Theta_y | \mathbf{y}, \mathcal{X})$ where Θ_y is the

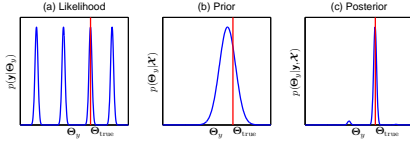


Figure 1. (a) The likelihood function. (b) The prior. (c) The posterior. This schematic diagram shows that an appropriate prior gives the desirable MAP solution. The vertical line shows the true value of Θ_y .

GM parameter set for \mathbf{y} . By Bayes' theorem, the posterior can be written in the form $p(\Theta_y|\mathbf{y}, \mathcal{X}) \propto p(\mathbf{y}|\Theta_y)p(\Theta_y|\mathcal{X})$.

The key issue of Bayesian SS is how to set up the prior $p(\Theta_y|\mathcal{X})$. If overlapping partials are present, the frequency matrix \mathbf{H} is rank deficient and many choices of Θ can give similar values of the likelihood $p(\mathbf{y}|\Theta_y)$. Hence, there are many peaks in the likelihood function as shown in the schematic diagram (Figure 1(a)). In order to find the desirable MAP solution, it is advantageous that the prior distribution has a high density around the correct value of Θ_y . In Figure 1(b), the prior is appropriate so that the MAP solution, i.e. the peak of the posterior, can be located correctly as depicted in Figure 1(c). In short, an appropriate prior of the GM parameters is crucial for resolving the overlapping partials. It can be found by using the training data and the estimated PM parameters.

The prior $p(\Theta_y|\mathcal{X})$ expresses the probability distribution of Θ_y given the training data \mathcal{X} and before the mixture \mathbf{y} is observed. The functional form of $p(\Theta_y|\mathcal{X})$ can be formulated in terms of PM. The PM parameter set Ψ_y of the mixture \mathbf{y} is divided into two sets: the invariant PM parameter set $\Psi_{y,\mathbb{I}}$ and the varying PM parameter set $\Psi_{y,\mathbb{V}}$. For the training data \mathcal{X} , the PM parameter set $\Psi_{\mathcal{X}}$ is divided into the invariant PM parameter set $\Psi_{\mathcal{X},\mathbb{I}}$ and the varying PM parameter set $\Psi_{\mathcal{X},\mathbb{V}}$. Note that both the mixture and the training data share the same set of the invariant PM parameters. The subscripts y and \mathcal{X} for the invariant PM parameters can be omitted for clarity so $\Psi_{\mathbb{I}} = \Psi_{y,\mathbb{I}} = \Psi_{\mathcal{X},\mathbb{I}}$.

The posterior $p(\Theta_y|\mathbf{y}, \mathcal{X})$ of the GM parameters can be linked up with the PM parameters by using marginalization:

$$p(\Theta_y|\mathbf{y}, \mathcal{X}) = \iint p(\Theta_y, \Psi_{y,\mathbb{V}}, \Psi_{\mathbb{I}}|\mathbf{y}, \mathcal{X}) d\Psi_{y,\mathbb{V}} d\Psi_{\mathbb{I}}. \quad (7)$$

Note that the noise variance σ_n^2 of the mixture in (4) is omitted in the derivation for clarity. Then by the product rule of probability, (7) can be put into

$$p(\Theta_y|\mathbf{y}, \mathcal{X}) = \iint p(\Theta_y|\mathbf{y}, \mathcal{X}, \Psi_{y,\mathbb{V}}, \Psi_{\mathbb{I}}) p(\Psi_{y,\mathbb{V}}, \Psi_{\mathbb{I}}|\mathbf{y}, \mathcal{X}) d\Psi_{y,\mathbb{V}} d\Psi_{\mathbb{I}} \quad (8)$$

where the first term is the posterior of Θ_y and the second is the posterior of $\Psi_{y,\mathbb{V}}$ and $\Psi_{\mathbb{I}}$.

However, finding the MAP solution involves evaluating the integration over all possible values of $\Psi_{y,\mathbb{V}}$ and $\Psi_{\mathbb{I}}$ in (8). PM is a highly dimensional and nonlinear model that makes the integration analytically infeasible. Different approximation techniques can be used to find the MAP solution.

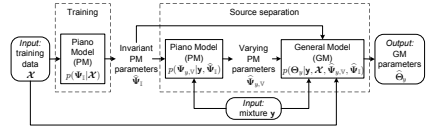


Figure 2. Bayesian framework for SS.

For computational efficiency, here we have used the evidence approximation [12, 13]. Following the derivation of the evidence approximation in [3, p. 408], we assume that the posterior $p(\Psi_{y,\mathbb{V}}, \Psi_{\mathbb{I}}|\mathbf{y}, \mathcal{X})$ is sharply peaked around their most probable values $\hat{\Psi}_{y,\mathbb{V}}$ and $\hat{\Psi}_{\mathbb{I}}$. Then (8) can be written as $p(\Theta_y|\mathbf{y}, \mathcal{X}) \approx p(\Theta_y|\mathbf{y}, \mathcal{X}, \hat{\Psi}_{y,\mathbb{V}}, \hat{\Psi}_{\mathbb{I}})$.

Hence, the MAP solution $\hat{\Theta}_y$ is the maximum of the posterior $p(\Theta_y|\mathbf{y}, \mathcal{X}, \hat{\Psi}_{y,\mathbb{V}}, \hat{\Psi}_{\mathbb{I}})$. The estimation of $\hat{\Psi}_{y,\mathbb{V}}$ and $\hat{\Psi}_{\mathbb{I}}$ can be done as follows: (i) $\hat{\Psi}_{y,\mathbb{V}}$ is estimated by maximizing the posterior $p(\Psi_{y,\mathbb{V}}|\mathbf{y}, \mathcal{X})$ via the evidence approximation which gives $p(\Psi_{y,\mathbb{V}}|\mathbf{y}, \mathcal{X}) \approx p(\Psi_{y,\mathbb{V}}|\mathbf{y}, \hat{\Psi}_{\mathbb{I}})$ (note that \mathcal{X} is omitted because $\Psi_{y,\mathbb{V}}$ is independent of \mathcal{X} if $\hat{\Psi}_{\mathbb{I}}$ is given); (ii) $\hat{\Psi}_{\mathbb{I}}$ is estimated by maximizing the posterior $p(\Psi_{\mathbb{I}}|\mathbf{y}, \mathcal{X})$ that can be approximated by using the training data only so $p(\Psi_{\mathbb{I}}|\mathbf{y}, \mathcal{X}) \approx p(\Psi_{\mathbb{I}}|\mathcal{X})$.

According to these results, the whole SS process is summarized in Figure 2. The whole process is divided into the following two stages:

1. *Training.* Given the training data \mathcal{X} , find the most probable value of the invariant PM parameters $\hat{\Psi}_{\mathbb{I}}$ of $p(\Psi_{\mathbb{I}}|\mathcal{X})$.
2. *SS.* Given the mixture \mathbf{y} , the training data \mathcal{X} and the invariant PM parameters $\hat{\Psi}_{\mathbb{I}}$, SS functions in two steps:
 - (a) *SS with PM.* Given \mathbf{y} and $\hat{\Psi}_{\mathbb{I}}$, find the most probable value of the varying PM parameters $\hat{\Psi}_{y,\mathbb{V}}$ of $p(\Psi_{y,\mathbb{V}}|\mathbf{y}, \hat{\Psi}_{\mathbb{I}})$.
 - (b) *SS with GM.* Given \mathbf{y} , \mathcal{X} , $\hat{\Psi}_{y,\mathbb{V}}$ and $\hat{\Psi}_{\mathbb{I}}$, find the MAP solution $\hat{\Theta}_y$ of $p(\Theta_y|\mathbf{y}, \mathcal{X}, \hat{\Psi}_{y,\mathbb{V}}, \hat{\Psi}_{\mathbb{I}})$.

4. TRAINING AND SS WITH PM

The goal of the training stage is to find the most probable invariant PM parameters $\hat{\Psi}_{\mathbb{I}}$ that maximize the posterior of the invariant PM parameters $p(\Psi_{\mathbb{I}}|\mathcal{X})$ given the training data \mathcal{X} . By Bayes' theorem, the posterior can be rewritten as $p(\Psi_{\mathbb{I}}|\mathcal{X}) \propto p(\mathcal{X}|\Psi_{\mathbb{I}})p(\Psi_{\mathbb{I}})$. The prior $p(\Psi_{\mathbb{I}})$ reflects our prior knowledge of the invariant PM parameters $\Psi_{\mathbb{I}}$. The values of $\Psi_{\mathbb{I}}$ greatly vary from different pitches and pianos. If we have little idea on suitable values for a parameter, it is safe to assign a prior which is insensitive to the values of that parameter [4]. Then maximizing the posterior $p(\Psi_{\mathbb{I}}|\mathcal{X})$ is effectively equivalent to maximize the likelihood $p(\mathcal{X}|\Psi_{\mathbb{I}})$. The details of finding the solution $\hat{\Psi}_{\mathbb{I}}$ can be found in [15].

Given the invariant PM parameters $\hat{\Psi}_{\mathbb{I}}$ and the mixture \mathbf{y} , we perform SS with PM as shown in Figure 2. The goal of SS

with PM is to find the most probable varying PM parameters $\hat{\Psi}_{y,\mathcal{V}}$ that maximize the posterior of the varying PM parameters $p(\Psi_{y,\mathcal{V}}|\mathbf{y}, \hat{\Psi}_{\mathbb{I}})$. By Bayes' theorem, the posterior can be rewritten as $p(\Psi_{y,\mathcal{V}}|\mathbf{y}, \hat{\Psi}_{\mathbb{I}}) \propto p(\mathbf{y}|\Psi_{y,\mathcal{V}}, \hat{\Psi}_{\mathbb{I}})p(\Psi_{y,\mathcal{V}})$. The prior $p(\Psi_{y,\mathcal{V}})$ reflects our prior knowledge of the invariant PM parameters $\Psi_{y,\mathcal{V}}$. The values of $\Psi_{y,\mathcal{V}}$ greatly vary from different playings. Hence, we choose an insensitive prior for $\Psi_{y,\mathcal{V}}$ as $\Psi_{\mathbb{I}}$. Then maximizing the posterior $p(\Psi_{y,\mathcal{V}}|\mathbf{y}, \hat{\Psi}_{\mathbb{I}})$ is again effectively equivalent to maximize the likelihood $p(\mathbf{y}|\Psi_{y,\mathcal{V}}, \hat{\Psi}_{\mathbb{I}})$. The details of finding $\hat{\Psi}_{y,\mathcal{V}}$ are also presented in [15].

5. SS WITH GM

The process of SS with GM is divided into the following two steps: (1) estimate the hyperparameters, and (2) given the hyperparameters, find the MAP solution $\hat{\Theta}_y$. We will focus on the second step first.

5.1 Find the MAP solution

The MAP solution $\hat{\Theta}_y$ is found by maximizing the posterior $p(\Theta_y|\mathbf{y}, \mathcal{X}, \hat{\Psi}_{y,\mathcal{V}}, \hat{\Psi}_{\mathbb{I}})$. The GM parameters Θ_y include the amplitude matrix \mathbf{G} and the frequencies \mathbf{f} . An iterative update scheme is designed to find the MAP solution: (1) given \mathbf{f} , update \mathbf{G} , and (2) given \mathbf{G} , update \mathbf{f} . Steps 1 to 2 are repeated until convergence. The iterative update starts with the input frequencies from the estimated frequencies in PM in Section 4. The frequencies in PM are close to those in GM. We find that 10 iterations are enough for convergence. In the followings, the iterative update scheme will be discussed in details.

5.1.1 Step 1: update the amplitude matrix \mathbf{G}

Each \mathbf{g}_r in the amplitude matrix \mathbf{G} can be estimated independently. Given the estimated frequencies $\hat{\mathbf{f}}_r$, now we rewrite the posterior of \mathbf{g}_r into $p(\mathbf{g}_r|\mathbf{y}_r, \mathcal{X}, \hat{\mathbf{f}}_r, \hat{\Psi}_{y,\mathcal{V}}, \hat{\Psi}_{\mathbb{I}}, \hat{\sigma}_{v_r}^2)$. The goal of this step is to find the MAP solution $\hat{\mathbf{g}}_r$ which maximizes the posterior of \mathbf{g}_r . By Bayes' theorem, the posterior of \mathbf{g}_r can be expressed in the form of

$$p(\mathbf{g}_r|\mathbf{y}_r, \mathcal{X}, \hat{\mathbf{f}}_r, \hat{\Psi}_{y,\mathcal{V}}, \hat{\Psi}_{\mathbb{I}}, \hat{\sigma}_{v_r}^2) \propto p(\mathbf{y}_r|\mathbf{g}_r, \hat{\mathbf{f}}_r, \hat{\sigma}_{v_r}^2)p(\mathbf{g}_r|\mathcal{X}, \hat{\Psi}_{y,\mathcal{V}}, \hat{\Psi}_{\mathbb{I}}) \quad (9)$$

where $\hat{\sigma}_{v_r}^2$ represents the estimated variance of the zero-mean Gaussian noise in (4).

The prior $p(\mathbf{g}_r|\mathcal{X}, \hat{\Psi}_{y,\mathcal{V}}, \hat{\Psi}_{\mathbb{I}})$ in (9) represents the prior distribution of \mathbf{g}_r conditioned on the training data \mathcal{X} and the PM parameters $\hat{\Psi}_{y,\mathcal{V}}$ and $\hat{\Psi}_{\mathbb{I}}$. It is modeled as a Gaussian with the mean $\hat{\boldsymbol{\mu}}_{g_r}$ and the covariance matrix $\hat{\boldsymbol{\Sigma}}_{g_r}$. In this section, it is assumed that the hyperparameters $\hat{\sigma}_{v_r}^2$, $\hat{\boldsymbol{\mu}}_{g_r}$, and $\hat{\boldsymbol{\Sigma}}_{g_r}$ have been estimated and their values are known. The estimation of these hyperparameters from \mathcal{X} , $\hat{\Psi}_{y,\mathcal{V}}$ and $\hat{\Psi}_{\mathbb{I}}$ will be discussed in Section 5.2. Note that each \mathbf{g}_r has its own set of $\hat{\boldsymbol{\mu}}_{g_r}$ and $\hat{\boldsymbol{\Sigma}}_{g_r}$ so the MAP solution of each \mathbf{g}_r can be found independently.

As $\tilde{\mathbf{y}}_r = \mathbf{H}\mathbf{g}_r$ is a linear model for given \mathbf{H} , and both the noise and the prior are Gaussian, the resulting posterior of \mathbf{g}_r is also Gaussian. Therefore, the MAP solution $\hat{\mathbf{g}}_r$ is equal to the posterior mean. By using the result in [4, p. 153], the MAP solution is

$$\hat{\mathbf{g}}_r = (\hat{\boldsymbol{\Sigma}}_{g_r}^{-1} + \hat{\sigma}_{v_r}^{-2}\mathbf{H}^T\mathbf{H})^{-1} (\hat{\boldsymbol{\Sigma}}_{g_r}^{-1}\hat{\boldsymbol{\mu}}_{g_r} + \hat{\sigma}_{v_r}^{-2}\mathbf{H}^T\mathbf{y}_r). \quad (10)$$

5.1.2 Step 2: update the frequencies \mathbf{f}

Given the estimated amplitude matrix $\hat{\mathbf{G}}$ in Step 1, the goal of Step 2 is to find the MAP solution $\hat{\mathbf{f}}$ which maximizes the posterior $p(\mathbf{f}|\mathbf{Y}, \hat{\mathbf{G}}, \hat{\Psi}_{y,\mathcal{V}}, \hat{\Psi}_{\mathbb{I}}, \hat{\sigma}_v^2)$. However, the model $\hat{\mathbf{Y}} = \mathbf{H}\mathbf{G}$ in (5) is nonlinear with \mathbf{f} . Based on our work in [14], we vectorize the matrix $\hat{\mathbf{Y}}$ into $\hat{\mathbf{Y}}_{\text{vec}}$ and then linearize $\hat{\mathbf{Y}}_{\text{vec}}$ by using Taylor's expansion so

$$\hat{\mathbf{Y}}_{\text{vec}}(\mathbf{f}) \approx \hat{\mathbf{Y}}_{\text{vec}}(\mathbf{f}^{\text{cur}}) + \mathbf{Z}(\mathbf{f}^{\text{cur}})(\mathbf{f} - \mathbf{f}^{\text{cur}}) \quad (11)$$

where $\hat{\mathbf{Y}}_{\text{vec}}(\mathbf{f})$ is the estimate depending on the new frequency vector \mathbf{f} which is to be updated, and $\hat{\mathbf{Y}}_{\text{vec}}(\mathbf{f}^{\text{cur}})$ is the estimate depending on the current estimate of \mathbf{f}^{cur} . The matrix $\mathbf{Z} = \mathbf{Z}(\mathbf{f}^{\text{cur}})$ is the Jacobian matrix $\partial\hat{\mathbf{Y}}_{\text{vec}}/\partial\mathbf{f}$ evaluated at \mathbf{f}^{cur} and $\mathbf{Z} = [\mathbf{Z}_1^T \dots \mathbf{Z}_r^T \dots \mathbf{Z}_R^T]^T$. The matrix \mathbf{Z}_r is the Jacobian matrix $\partial\hat{\mathbf{y}}_r/\partial\mathbf{f}$ at r th frame for all tones and $\mathbf{Z}_r = [\mathbf{Z}_{1,r} \dots \mathbf{Z}_{k,r} \dots \mathbf{Z}_{K,r}]$. Then the Jacobian matrix $\mathbf{Z}_{k,r}$ at r th frame for k th tone is

$$\mathbf{Z}_{k,r}[l, m] = \frac{\partial \hat{y}_r[l]}{\partial f_{k,m}} = 2\pi l m w[l] (-\alpha_{k,m,r} \sin(2\pi f_{k,m} t_l) + \beta_{k,m,r} \cos(2\pi f_{k,m} t_l)). \quad (12)$$

Hence, each element in \mathbf{Z} can be computed from (12).

Following the prior distribution of \mathbf{g}_r , the prior distribution of \mathbf{f} is also modeled as a Gaussian with the mean $\hat{\boldsymbol{\mu}}_f$ and the covariance matrix $\hat{\boldsymbol{\Sigma}}_f$. By applying (11) to the result in [4, p. 93], the MAP solution $\hat{\mathbf{f}}$ is

$$\hat{\mathbf{f}} = (\hat{\boldsymbol{\Sigma}}_f^{-1} + \mathbf{Z}^T \hat{\boldsymbol{\Sigma}}_v^{-1} \mathbf{Z})^{-1} (\hat{\boldsymbol{\Sigma}}_f^{-1} \hat{\boldsymbol{\mu}}_f + \mathbf{Z}^T \hat{\boldsymbol{\Sigma}}_v^{-1} (\hat{\mathbf{Y}}_{\text{vec}} - \hat{\mathbf{Y}}_{\text{vec}}(\mathbf{f}^{\text{cur}}) + \mathbf{Z}\mathbf{f}^{\text{cur}})) \quad (13)$$

where $\mathbf{Z} = \mathbf{Z}(\mathbf{f}^{\text{cur}})$, $\hat{\mathbf{Y}}_{\text{vec}} = \hat{\mathbf{Y}}_{\text{vec}}(\mathbf{f}^{\text{cur}})$, and the covariance matrix $\hat{\boldsymbol{\Sigma}}_v = \text{diag}(\hat{\sigma}_{v_1}^2 \mathbf{1}_L, \dots, \hat{\sigma}_{v_R}^2 \mathbf{1}_L)$ and $\mathbf{1}_L$ denotes the L -dimensional column vector filled with 1's. In the next section, we will show how to find the hyperparameters which are crucial for resolving overlapping partials.

5.2 Estimation of the hyperparameters

Given the training data \mathcal{X} , we first estimate the GM parameters for each isolated tone in \mathcal{X} by the method in [14]. Together with the estimated PM parameters $\hat{\Psi}_{y,\mathcal{V}}$ and $\hat{\Psi}_{\mathbb{I}}$ found in Section 4, we will estimate the hyperparameters $\hat{\sigma}_{v_r}^2$, $\hat{\boldsymbol{\mu}}_{g_r}$, $\hat{\boldsymbol{\Sigma}}_{g_r}$, $\hat{\boldsymbol{\mu}}_f$ and $\hat{\boldsymbol{\Sigma}}_f$.

5.2.1 Estimation of the noise variance $\hat{\sigma}_{v_r}^2$

To estimate the noise variance $\hat{\sigma}_{v_r}^2$ of \mathbf{y}_r in (4), we model the noise variance of an isolated tone $\mathbf{x}_{k,r}$ at a frame is directly proportional to the signal power. Then the noise variance of $\mathbf{x}_{k,r}$ is $\hat{\sigma}_{v_{k,r}}^2 = \hat{\sigma}_{v_k}^2 \|\mathbf{x}_{k,r}\|^2$ where $\hat{\sigma}_{v_k}^2$ is the proportionality constant for pitch p_k and it can be determined by the training data \mathcal{X} which may contain multiple instances of the same pitch. Let $\mathbf{x}_{k,r,\mathcal{X}}^i$ be a frame of an isolated tone in \mathcal{X} where the index i denotes the i th instance of the pitch p_k . Then $\hat{\sigma}_{v_k}^2$ can be estimated by

$$\hat{\sigma}_{v_k}^2 = \frac{1}{I_k R_k L} \sum_{i=1}^{I_k} \sum_{r=1}^{R_k} \sum_{l=0}^{L-1} \left(\frac{x_{k,r,\mathcal{X}}^i[l] - \hat{x}_{k,r,\mathcal{X}}^i[l]}{\|\mathbf{x}_{k,r,\mathcal{X}}^i\|} \right)^2 \quad (14)$$

where I_k is the number of instances of pitch p_k in \mathcal{X} , R_k^i is the number of frames in the i th instance of the pitch p_k and $R_k = \sum_{i=1}^{I_k} R_k^i$ and $\hat{\mathbf{x}}_{k,r,\mathcal{X}}$ is the estimate of $\mathbf{x}_{k,r,\mathcal{X}}$ and is found by using the method in [14].

The noise variance of the mixture \mathbf{y}_r is

$$\sigma_{v_r}^2 = \sum_{k=1}^K \sigma_{v_{k,r,y}}^2 = \sum_{k=1}^K \sigma_{v_k}^2 \|\mathbf{x}_{k,r,y}\|^2 \quad (15)$$

where $\mathbf{x}_{k,r,y}$ is the k th individual tone in the mixture, and $\sigma_{v_{k,r,y}}^2$ is its noise variance. However, $\mathbf{x}_{k,r,y}$ is not known. In order to estimate $\sigma_{v_r}^2$, we approximate $\|\mathbf{x}_{k,r,y}\|^2$ into

$$\|\mathbf{x}_{k,r,y}\|^2 \approx \left(\frac{\hat{c}_k}{\sum_{k=1}^K \hat{c}_k} \right) \|\mathbf{y}_r\|^2 \quad (16)$$

where the estimated intensity \hat{c}_k in PM determines the proportion of $\|\mathbf{x}_{k,r,y}\|^2$ in $\|\mathbf{y}_r\|^2$. Substituting (16) into (15), we estimate the noise variance $\hat{\sigma}_{v_r}^2$ in the mixture \mathbf{y}_r in the form of

$$\hat{\sigma}_{v_r}^2 = \sum_{k=1}^K \left(\frac{\hat{c}_k \hat{\sigma}_{v_k}^2}{\sum_{k=1}^K \hat{c}_k} \right) \|\mathbf{y}_r\|^2. \quad (17)$$

5.2.2 Estimation of the prior distribution of the amplitudes \mathbf{g}_r

The prior distribution $p(\mathbf{g}_r | \hat{\boldsymbol{\mu}}_{g_r}, \hat{\boldsymbol{\Sigma}}_{g_r})$ of \mathbf{g}_r is modeled as the Gaussian with the mean $\hat{\boldsymbol{\mu}}_{g_r}$ and the covariance $\hat{\boldsymbol{\Sigma}}_{g_r}$. Both $\hat{\boldsymbol{\mu}}_{g_r}$ and $\hat{\boldsymbol{\Sigma}}_{g_r}$ depend on $\hat{\boldsymbol{\Psi}}_{y,\mathcal{V}}$ and $\hat{\boldsymbol{\Psi}}_{\mathbb{I}}$. This dependence can be formulated by converting the PM parameters $\hat{\boldsymbol{\Psi}}_{y,\mathcal{V}}$ and $\hat{\boldsymbol{\Psi}}_{\mathbb{I}}$ into the GM parameters. Let t'_r be the time at the center of the r th frame so that $t'_r = ((r-1)D + 0.5L)/f_s$ where D is the hop size in samples. Evaluating the envelope function of PM in (6) at the center of the r th frame, we can find the estimated amplitude $\hat{a}_{k,m,r,y,\text{PM}} = a(t'_r; \hat{c}_k, \hat{\varphi}_m)$ where \hat{c}_k and $\hat{\varphi}_m$ are included in $\hat{\boldsymbol{\Psi}}_{y,\mathcal{V}}$ and $\hat{\boldsymbol{\Psi}}_{\mathbb{I}}$ respectively.

The phase at the center of r th frame can be calculated from $\hat{\boldsymbol{\Psi}}_{y,\mathcal{V}}$ and $\hat{\boldsymbol{\Psi}}_{\mathbb{I}}$ by

$$\hat{\phi}_{k,m,r,y,\text{PM}} = 2\pi \hat{f}_{k,m,\text{PM}}(t'_r - \hat{\tau}_k) + \hat{\phi}_{k,m,\text{PM}} \quad (18)$$

where the frequency $\hat{f}_{k,m,y,\text{PM}}$ and the phase $\hat{\phi}_{k,m,\text{PM}}$ are included in $\hat{\boldsymbol{\Psi}}_{\mathbb{I}}$, and the time shift $\hat{\tau}_k$ is included in $\hat{\boldsymbol{\Psi}}_{y,\mathcal{V}}$. Then $\hat{a}_{k,m,r,y,\text{PM}}$ and $\hat{\phi}_{k,m,r,y,\text{PM}}$ in PM can be transformed into the amplitude of cosine $\hat{\alpha}_{k,m,r,y,\text{PM}}$ and the amplitude of sine $\hat{\beta}_{k,m,r,y,\text{PM}}$ in GM. The mean $\hat{\boldsymbol{\mu}}_{g_r}$ of the prior is assigned to be these estimated amplitudes from PM so that

$$\hat{\boldsymbol{\mu}}_{\alpha_{k,m,r}} = \hat{\alpha}_{k,m,r,y,\text{PM}} = \hat{a}_{k,m,r,y,\text{PM}} \cos \hat{\phi}_{k,m,r,y,\text{PM}} \quad (19)$$

$$\hat{\boldsymbol{\mu}}_{\beta_{k,m,r}} = \hat{\beta}_{k,m,r,y,\text{PM}} = -\hat{a}_{k,m,r,y,\text{PM}} \sin \hat{\phi}_{k,m,r,y,\text{PM}} \quad (20)$$

where $\hat{\boldsymbol{\mu}}_{\alpha_{k,m,r}}$ and $\hat{\boldsymbol{\mu}}_{\beta_{k,m,r}}$ are the elements in $\hat{\boldsymbol{\mu}}_{g_r}$ and they follow the ordering in (3).

The covariance $\hat{\boldsymbol{\Sigma}}_{g_r}$ measures the deviation between the values of \mathbf{g}_r , estimated by PM and those estimated by GM. It is modeled as a diagonal matrix of which the diagonal is filled with the variances $\hat{\sigma}_{\alpha_{k,m,r}}^2$ and $\hat{\sigma}_{\beta_{k,m,r}}^2$ and follows the ordering in (3). We assume that the variances $\hat{\sigma}_{\alpha_{k,m,r}}^2$ and $\hat{\sigma}_{\beta_{k,m,r}}^2$ are identical and they are directly proportional to the power of the partial amplitude. This gives

$$\hat{\sigma}_{\alpha_{k,m,r}}^2 = \hat{\sigma}_{\beta_{k,m,r}}^2 = \bar{\sigma}_{G_k}^2 (\hat{a}_{k,m,r,y,\text{PM}})^2 \quad (21)$$

where $\bar{\sigma}_{G_k}^2$ is the proportionality constant and it can be determined by the training data \mathcal{X} as below.

Let $\hat{\alpha}_{k,m,r,\mathcal{X},\text{GM}}^i$ and $\hat{\beta}_{k,m,r,\mathcal{X},\text{GM}}^i$ be the amplitudes in GM for \mathcal{X} and they have been estimated by the method in [14]. Let $\hat{\alpha}_{k,m,r,\mathcal{X},\text{PM}}^i$ and $\hat{\beta}_{k,m,r,\mathcal{X},\text{PM}}^i$ be the amplitudes in GM for \mathcal{X} and they are converted from the PM estimate. The conversion from the PM estimate to the GM estimate for \mathcal{X} follows that for the mixture \mathbf{y} in (19) and (20). Let $\hat{a}_{k,m,r,\mathcal{X},\text{PM}}^i$ be the partial amplitude in PM then

$$\hat{a}_{k,m,r,\mathcal{X},\text{PM}}^i = \sqrt{(\hat{\alpha}_{k,m,r,\mathcal{X},\text{GM}}^i)^2 + (\hat{\beta}_{k,m,r,\mathcal{X},\text{GM}}^i)^2}. \quad (22)$$

Following (21), we can estimate $\bar{\sigma}_{G_k}^2$ from \mathcal{X} by

$$\bar{\sigma}_{G_k}^2 = \frac{1}{2I_k M_k R_k} \sum_{i=1}^{I_k} \sum_{m=1}^{M_k} \sum_{r=1}^{R_k} \left\{ \left(\frac{\delta \hat{\alpha}_{k,m,r}^i}{\hat{a}_{k,m,r,\mathcal{X},\text{PM}}^i} \right)^2 + \left(\frac{\delta \hat{\beta}_{k,m,r}^i}{\hat{a}_{k,m,r,\mathcal{X},\text{PM}}^i} \right)^2 \right\} \quad (23)$$

where $\delta \hat{\alpha}_{k,m,r}^i = \hat{\alpha}_{k,m,r,\mathcal{X},\text{GM}}^i - \hat{\alpha}_{k,m,r,\mathcal{X},\text{PM}}^i$ and $\delta \hat{\beta}_{k,m,r}^i = \hat{\beta}_{k,m,r,\mathcal{X},\text{GM}}^i - \hat{\beta}_{k,m,r,\mathcal{X},\text{PM}}^i$.

Note that the prior $p(\mathbf{g}_r | \hat{\boldsymbol{\mu}}_{g_r}, \hat{\boldsymbol{\Sigma}}_{g_r})$ reflects the difference between the individual tones estimated by GM and PM. As PM gives satisfactory quality of estimation in [15], the difference should be small enough to make the prior distribution $p(\mathbf{g}_r | \hat{\boldsymbol{\mu}}_{g_r}, \hat{\boldsymbol{\Sigma}}_{g_r})$ has a high density around the correct value of \mathbf{g}_r as shown in the schematic diagram in Figure 1. Hence, overlapping partials can be resolved and higher quality of SS can be obtained. It will be verified and explained in the experiments.

5.2.3 Estimation of the prior distribution of frequencies \mathbf{f}

The prior distribution $p(\mathbf{f} | \hat{\boldsymbol{\mu}}_f, \hat{\boldsymbol{\Sigma}}_f)$ of \mathbf{f} is modeled as the Gaussian with the mean $\hat{\boldsymbol{\mu}}_f$ and the covariance $\hat{\boldsymbol{\Sigma}}_f$. The mean $\hat{\boldsymbol{\mu}}_f$ is set to the estimated frequencies in PM from $\hat{\boldsymbol{\Psi}}_{\mathbb{I}}$ so that

$$\hat{\boldsymbol{\mu}}_{f_{k,m}} = \hat{f}_{k,m,\text{PM}} \quad (24)$$

where $\hat{\boldsymbol{\mu}}_{f_{k,m}}$ are the elements in $\hat{\boldsymbol{\mu}}_f$. Following the derivation of $\hat{\boldsymbol{\Sigma}}_{g_r}$, we also assume that $\hat{\boldsymbol{\Sigma}}_f$ is a diagonal matrix of which the diagonal is filled with each variance $\hat{\sigma}_{f_{k,m}}^2$. The variance $\hat{\sigma}_{f_{k,m}}^2$ is modeled to be directly proportional to the square of the frequency in PM. This gives

$$\hat{\sigma}_{f_{k,m}}^2 = \bar{\sigma}_{f_k}^2 (\hat{f}_{k,m,\text{PM}})^2 \quad (25)$$

where $\bar{\sigma}_{f_k}^2$ is the proportionality constant which can also be determined by the training data \mathcal{X} . The estimate of $\bar{\sigma}_{f_k}^2$ is

$$\bar{\sigma}_{f_k}^2 = \frac{1}{M_k} \sum_{m=1}^{M_k} \left(\frac{\hat{f}_{k,m,\mathcal{X},\text{GM}} - \hat{f}_{k,m,\text{PM}}}{\hat{f}_{k,m,\text{PM}}} \right)^2 \quad (26)$$

where $\hat{f}_{k,m,\mathcal{X},\text{GM}}$ is the estimated frequency in GM for \mathcal{X} and it can be estimated by using the method in [14]. Note that there is no subscript \mathcal{X} in $\hat{f}_{k,m,\text{PM}}$ because $\hat{f}_{k,m,\text{PM}}$ are the invariant PM parameters so the training data and the mixture share the same set of $\hat{f}_{k,m,\text{PM}}$.

In summary, after estimating the hyperparameters $\hat{\sigma}_{v_r}^2$ in (17), $\hat{\boldsymbol{\mu}}_{g_r}$ in (19) and (20), $\hat{\boldsymbol{\Sigma}}_{g_r}$ in (21), $\hat{\boldsymbol{\mu}}_f$ in (24) and $\hat{\boldsymbol{\Sigma}}_f$ in (25), we can find the MAP solution $\hat{\Theta}_{g_r}$ of GM by iteratively updating the amplitude matrix \mathbf{G} in (10) and the frequencies \mathbf{f} in (13). In the next section, experimental results will be

presented to show the performance of the whole SS process.

6. EXPERIMENTS

6.1 Data set and experimental setup

We used the same data set in [15] for comparing the performance. The data set contains 25 mixtures. Each mixture was generated by mixing the isolated tones in the recorded piano databases [8, 15], taken from 4 different pianos. Only tones from the same piano were used to form a mixture. The pitches in each mixture correspond to a chord randomly selected from 11 piano pieces in the RWC database [8]. The number of tones (represented by K) in our selected mixtures ranges from 1 to 6: 1 tone (8 mixtures), 2 tones (6), 3 tones (5), 4 tones (4), 5 tones (1) and 6 tones (1). These 25 mixtures consist of 62 tones. 7 mixtures contain one pair of octaves, 2 ($K = 5$ and $K = 6$) contain 2 pairs of octaves. For the training data, two instances of each pitch are available so $I_k = 2$. The first 0.5 second of the mixtures and the training data were used in the experiments. All data were downsampled to 11.025 kHz for faster processing. The window setting in GM is as follows: the window function is the hamming window with length 11.61 ms ($L = 128$) and 50% overlap. The titles of the piano pieces used and details of the selected mixtures are available on the website of this paper (link available at the end of this section).

6.2 Results

The performance of our SS system is evaluated by the signal-to-noise ratio (SNR) defined by

$$\text{SNR} = 10 \log_{10} \frac{\sum_n x(t_n)^2}{\sum_n (x(t_n) - \hat{x}(t_n))^2} \quad (27)$$

where $x(t_n)$ is the isolated tone in the time domain before mixing and $\hat{x}(t_n)$ is the estimated tone in the time domain. The isolated tones give the ground truth for evaluation.

6.2.1 Evaluation on modeling quality

We followed the procedures in [15] to evaluate the modeling quality, i.e. the quality of PM and GM to represent an isolated tone before mixing. The isolated tones of the 25 mixtures were inputted into our proposed SS system including both PM and GM. The outputs of our system were the estimated tones reconstructed from PM and GM. If the parameters obtained in PM and GM are accurate, they can regenerate the original tones in high quality. The result is that the average SNRs of PM and GM are 11.15 dB and 17.38 dB respectively. The average SNR of GM is much higher than that of PM. This is because GM is more flexible to represent piano tones.

6.2.2 Comparing with other systems for separation quality

The procedures in [15] were followed to evaluate the separation quality, i.e. the quality of PM and GM separating a mixture into its individual tones. We also compared PM and GM with a recent SS system in [11], in which Li, Woodruff and Wang built their system (Li's system) based on CAM mentioned in Section 1. It uses the non-overlapping partials to estimate the overlapping partials of the same note. The implementation of

	SNR (dB)		
	PM	GM	Li
All mixtures	10.88	13.51	6.63
$K=2$	11.76	15.26	12.07
$2 \leq K \leq 6$	10.97	13.15	5.40
Upper tones in octaves	10.95	12.77	1.57

Table 1. Comparison of Li's system and our PM and GM.

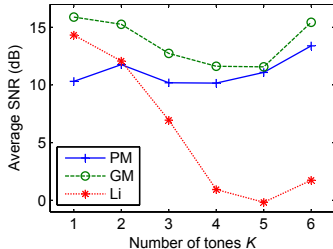


Figure 3. Average SNR against the number of tones K for PM, GM, and Li's system.

Li's system was provided by the authors. The true fundamental frequency of each tone was supplied to Li's system.

The results are shown in Table 1. For the 25 mixtures, the average SNRs of PM, GM and Li's system are 10.88 dB, 13.51 dB and 6.63 dB respectively. Both PM and GM outperform Li's system. A significant improvement is in the octave cases as shown in the table. Li's system is unable to resolve the overlapping partials of the upper tones in octaves because non-overlapping partials are not available. On the other hand, both PM and GM are able to reconstruct the upper tone in an octave. The overlapping partials were successfully resolved even for mixtures containing 2 pairs of octaves of C3, G3, C4, E4, G4 ($K = 5$) and of F#3, C4, F4, C5, D5, F5 ($K = 6$).

The average SNR against the number of tones K is plotted in Figure 3. The average SNR of Li's system decreases much more rapidly than PM and GM. Our system can make use of the training data to give higher separation quality. Some audio files in the experiments are selected for demonstration purpose. The audio files, titles of piano pieces used, details of the selected mixtures and mathematical notations used in this paper are available at <http://www.cse.cuhk.edu.hk/~khwong/www2/conference/ismir2015/ismir2015.html>.

7. CONCLUSIONS

Here we have proposed a score-informed monaural SS system to extract each tone from a mixture of piano tone signals. Two sinusoidal models, PM and GM, are employed to represent piano tones in the system. We formulate a hierarchical Bayesian framework to run both Models in the SS process so that the mixtures with overlapping partials can be resolved with high quality. Experiments show that our proposed system gives robust and accurate separations of mixtures and improves the separation quality significantly comparing to the previous work.

8. REFERENCES

- [1] A. Askenfelt, editor. *Five Lectures on the Acoustics of the Piano*. Royal Swedish Academy of Music, 1990. Available online at http://www.speech.kth.se/music/5_lectures/.
- [2] T. Berg-Kirkpatrick, J. Andreas, and D. Klein. Unsupervised transcription of piano music. In *Advances in Neural Information Processing Systems*, pages 1538–1546, 2014.
- [3] C. M. Bishop. *Neural Network for Pattern Recognition*. Oxford University Press, New York, 1995.
- [4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, New York, 2006.
- [5] E. Schubert D. Fabian, R. Timmers, editor. *Expressiveness in Music Performance: Empirical Approaches Across Styles and Cultures*. Oxford University Press, 2014.
- [6] M. Davy, S. Godsill, and J. Idier. Bayesian analysis of polyphonic western tonal music. *Journal of the Acoustical Society of America*, 119(4):2498–2517, April 2006.
- [7] M. R. Every and J. E. Szymanski. Separation of synchronous pitched notes by spectral filtering of harmonics. *IEEE Transactions on Audio, Speech & Language Processing*, 14(5):1845–1856, 2006.
- [8] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. Rwc music database: Music genre database and musical instrument sound database. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR 2003)*, pages 229–230, October 2003.
- [9] Jinyu Han and B. Pardo. Reconstructing completely overlapped notes from musical mixtures. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, pages 249–252, 2011.
- [10] A. Klapuri and M. Davy, editors. *Signal Processing Methods for Music Transcription*. Springer, 2006.
- [11] Y. Li, J. Woodruff, and D. Wang. Monaural musical sound separation based on pitch and common amplitude modulation. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(7):1361–1371, 2009.
- [12] D. J. C. MacKay. Bayesian interpolation. *Neural Computation*, 4:415–447, 1992.
- [13] D. J. C. MacKay. A practical bayesian framework for backpropagation networks. *Neural Computation*, 4:448–472, 1992.
- [14] W. M. Szeto and K. H. Wong. Sinusoidal modeling for piano tones. In *2013 IEEE International Conference on Signal Processing, Communication and Computing (ICSPCC)*, pages 1–6, Kunming, China, August 2013. Available online at <http://www.cse.cuhk.edu.hk/~khwong/www2/conference/ismir2015/ismir2015.html>.
- [15] W. M. Szeto and K. H. Wong. Source separation and analysis of piano music signals using instrument-specific sinusoidal model. In *Proceedings of the 16th International Conference on Digital Audio Effects (DAFx-13)*, pages 109–116, Maynooth, Ireland, September 2013. Available online at <http://www.cse.cuhk.edu.hk/~khwong/www2/conference/ismir2015/ismir2015.html>.
- [16] T. Virtanen. *Sound Source Separation in Monaural Music Signals*. PhD thesis, Tampere University of Technology, Finland, November 2006.
- [17] M. Zivanovic. Harmonic bandwidth companding for separation of overlapping harmonics in pitched signals. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(5):898–908, May 2015.
- [18] U. Zöler, editor. *DAFX - Digital Audio Effects*. Wiley, 2nd edition, 2011.

AUTOMATIC TUNE FAMILY IDENTIFICATION BY MUSICAL SEQUENCE ALIGNMENT

Patrick E. Savage

Tokyo University of the Arts, Dept. of Musicology
patsavagenz@gmail.com

Quentin D. Atkinson

Auckland University, Dept. of Psychology
q.atkinson@auckland.ac.nz

ABSTRACT

Musics, like languages and genes, evolve through a process of transmission, variation, and selection. Evolution of musical tune families has been studied qualitatively for over a century, but quantitative analysis has been hampered by an inability to objectively distinguish between musical similarities that are due to chance and those that are due to descent from a common ancestor. Here we propose an automated method to identify tune families by adapting genetic sequence alignment algorithms designed for automatic identification and alignment of protein families. We tested the effectiveness of our method against a high-quality ground-truth dataset of 26 folk tunes from four diverse tune families (two English, two Japanese) that had previously been identified and aligned manually by expert musicologists. We tested different combinations of parameters related to sequence alignment and to modeling of pitch, rhythm, and text to find the combination that best matched the ground-truth classifications. The best-performing automated model correctly grouped 100% (26/26) of the tunes in terms of overall similarity to other tunes, identifying 85% (22/26) of these tunes as forming distinct tune families. The success of our approach on a diverse, cross-cultural ground-truth dataset suggests promise for future automated reconstruction of musical evolution on a wide scale.

1. INTRODUCTION

Darwin's theory of evolution is a broad one that applies not only to biology but also to cultural forms such as language and music [21], [27]. Musicologists have long been interested in understanding how and why music evolves, particularly the three key mechanisms of 1) *transmission* between generations, 2) generation of musical *variation*, and 3) *selection* of certain variants over others [10], [21]. In some cases, historical notations, audio recordings, or other musical "fossils" allow us to document music's cultural evolution through the accumulation of minute variations over time [5], [14], [28]. More often, the process of oral transmission results in contemporaneous groups of related melodies known as "tune families" [2], careful

comparison of which can be used to partially reconstruct the process of musical evolution [4]. This situation is analogous to the evolution of language families and biological species [1].

Traditionally, analysis of tune family evolution has been done by manually identifying and aligning small groups of related melodies (see Fig. 1a) and then qualitatively comparing the similarities and differences. This led to two major challenges that limited the scale of tune family research: 1) the need for an automated method of comparing large numbers of melodies; and 2) the need for an objective means of determining tune family membership.

Thanks to the rise of music information retrieval (MIR), the first challenge has been largely overcome by automated sequence alignment algorithms for identifying melodic similarity [9], [16], [23], some of which have been specifically designed for studying tune families [24-26]. However, the second challenge remains unsolved, with tune family identification considered "currently too ambitious to perform automatically" [24].

Here we propose a novel method of tune family identification inspired by molecular genetics [8]. In particular, the problem of protein family identification shares many analogies with tune family identification. Proteins are biological molecules that are constructed by joining sequences of amino acids into 3-dimensional structures that function to catalyze biochemical reactions. Meanwhile, tunes are constructed by joining sequences of notes into multidimensional melodies that function to carry song lyrics, accompany dance, etc. When attempting to identify both protein families and tune families, a major challenge is to determine whether any observed similarities are due to chance or common ancestry.

We sought to develop automated methods for identifying and aligning tune families that could be used in future large-scale studies of musical evolution throughout the world. To do this, we adapted methods designed for identifying and aligning protein families and tested their effectiveness on a cross-cultural ground-truth set of well-established tune families that had already been manually identified and aligned by expert musicologists. We then tested out different model parameters to determine which parameters are most effective at capturing the known ground-truth patterns.

2. DATA

Our ground-truth dataset consisted of 26 melodies from four contrasting tune families that had previously been



© Patrick E. Savage, Quentin D. Atkinson.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Patrick E. Savage, Quentin D. Atkinson. "Automatic tune family identification by musical sequence alignment", 16th International Society for Music Information Retrieval Conference, 2015.



Figure 1. A sample portion of a manually aligned tune family. a) The opening phrase of three tunes manually aligned by Bayard [3] and identified as part of the tune family he labeled “Brave Donnelly”. b) The same information encoded as aligned pitch-class sequences using our proposed method (see Methods and Fig. 2). Note that keys are transposed so that the tonic (originally F) is always represented as C.

identified and aligned manually by expert musicologists¹. Two of these tune families were British-American tune families that had been chosen by Samuel Bayard (who coined the term “tune family”) in order to capture “...all the problems attending a comparative tune study, and all the important features of traditional development that we constantly encounter when we try to elucidate the really extensive families of tunes.” [3]. The other two were Japanese tune families chosen for similar reasons by the Japanese folksong scholars MACHIDA Kashō and TAKEUCHI Tsutomu [12]. We chose this dataset because we needed a known baseline against which to compare the effectiveness of our methods, and because we wanted our method to have cross-cultural validity that is not limited to idiosyncracies of the types of European-American folk tunes that have traditionally been studied. In addition, the first author has first-hand experience singing English and Japanese folksongs, and this dataset is also comparable to similar but larger collections of British-American and Japanese folk songs (approximately 5,000 each in [5], [18]) to which we aim to eventually apply these automated methods.

Music is much more than notes transcribed in a score. However, in order to understand tune family evolution, we need a standardized method of comparing tunes across time and space. To allow for analysis of tunes

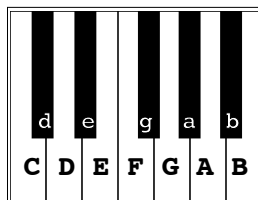


Figure 2. The most widely used “alphabet” for describing musical pitches divides an octave into 12 equally spaced semitones. Here these are visualized using the standard piano keyboard representation, with C representing the tonic.

documented before the advent of audio recording technology, this requires the use of transcriptions, although this comes at the cost of losing details about performance style (e.g., timbre, ornamentation, microtuning, microtiming). Furthermore, to allow evolutionary analysis using state-of-the-art methods from evolutionary biology, we need to further reduce the information in the score into aligned sequences. This approach was already implicit in the melodic alignment approach developed by tune family scholars, in which tunes were transposed into a common key and time signatures, phrases, and rhythms were stretched and compressed as necessary to align notes sharing similar pitches (see Fig. 1a).

Just as DNA can be modeled as a sequence constructed from an “alphabet” of 4 nucleic acids (C, G, A, or T) or a protein can be modeled as a sequence constructed from an alphabet of 20 amino acids, a melody can be modeled as a sequence constructed from an alphabet of 12 pitch classes representing the 12 notes of the chromatic scale (Fig. 2). By aligning sequences known to share common ancestry (as done manually in [3] and [12]), we can identify points on the alignment that are conserved, where a different pitch has been substituted, or where a pitch has been inserted/deleted (“indel”, represented using dashes). Fig. 1b shows how this method is used to encode the manual alignment shown in Fig. 1a. This information can then be analyzed quantitatively to reconstruct a phylogenetic tree, network, or other representation of the evolutionary history of the tune family.

The intuition of early tune family scholars to emphasize alignment of pitches, rather than rhythms or global stylistic features, is supported by recent research that has demonstrated quantitatively that pitch is greatly superior to rhythm and to global stylistic features both for the purposes of tune family identification in particular and for melodic similarity in general [23], [25]. However, judicious use of rhythm and other non-pitch features may improve tune family identification [25], and we explore this using several modeling techniques.

3. METHODS

3.1 Sequence alignment parameters

¹ Full metadata and aligned sequences are available at <http://dx.doi.org/10.6084/m9.figshare.1468015>

Automated sequence alignment requires a number of parameters to be defined. The choice of values for these parameters depends on the nature of the data and the goals of classification. Because automated tune family identification remains largely unexplored, we don't yet know which values are most appropriate for this goal. Therefore, we tested several values for each parameter to allow for empirical comparison of which parameter values performed best. When possible, we tested values that have worked well in similar work on protein family identification and automated melodic similarity algorithms.

3.1.1 Gap penalties

The functional mechanisms of protein structure result in substitutions being much more common than indels (insertions/deletions). Thus, most amino acid alignment algorithms set a gap opening penalty (GOP) parameter to be quite high to penalize the creation of gaps in a sequence. However, when indels do occur, they often encompass not only one amino acid residue, but rather can include fairly long sections. Thus gap extension penalties (GEP) are usually set to be substantially smaller than gap opening penalties (the default values for the popular ClustalW algorithm are for GOP and GEP values of 15 and 6.66, respectively [22]).

The mechanisms of musical sequence evolution are less well known, but previous tune family research suggests that insertion/deletion (e.g., of ornamentation) is quite common and may even be more common than substitution of different pitches. Thus, it seemed desirable to examine the effect of using a range of GOP and GEP values, ranging from the combination of GOP=0.8, GEP=0.2 used to align tunes in [25], to the amino acid alignment values given above. To do this, we chose GOP values of .8, 4, 8, 12, and 16, for each of which we tested GOP:GEP ratios of both 2 and 4. Thus, the gap penalty parameters ranged from minimums of GOP=0.8, GEP=0.2 (GOP:GEP ratio=4) to maximums of GOP=16, GEP=8 (GOP:GEP ratio=2). For all gap penalty parameters we followed previous tune family research [25] in using the Needleman-Wunsch alignment algorithm [17], as implemented in the *Biostrings* package in R V3.1.1 [19].

3.1.2 Pitch

There are various possibilities for weighting pitches to accommodate different degrees of similarity beyond simple match and mismatch. Previous weighting schemes using interval consonance or interval size have shown minimal improvement over a simple match/mismatch model [25]. Here we instead explore a novel weighting scheme based on qualitative tune family research that has found that tunes will sometimes change mode (i.e., some or all scale degrees may become flattened or sharpened to shift from major to minor or vice-versa [3]). To do this, we simply treated an alignment of major and minor versions of each scale degree as a match (i.e., treating lowercase letters in Fig. 2 as capitals).

3.1.3 Rhythm/text

Previous tune family research has suggested that some notes are likely to be more evolutionarily stable than others. In particular, notes that are rhythmically accented [6] or that carry text [11] are proposed to be more reliable in identifying tune families than rhythmically unaccented or non-text-carrying notes, respectively. To examine these possibilities, we contrasted the results using the full sequences with those using shorter sequences created by excluding rhythmically unaccented notes (i.e., notes not falling on the first beat of a measure) or non-text-carrying notes (e.g., notes where the vowel is held over from a previous note) from the full sequences.

3.1.4 Summary

In sum, we tested all possible combinations of the following parameters:

- 1) Gap opening penalty: i) .8, ii) 4, iii) 8, iv) 12 or v) 16
- 2) Gap opening penalty : Gap extension penalty (GOP:GEP) ratio: i) 2 or ii) 4
- 3) Pitch: i) including or ii) ignoring mode
- 4) Rhythm: i) including or ii) ignoring rhythmically unaccented notes
- 5) Text: i) including or ii) ignoring non-text-carrying notes

This gave a total of $5 \times 2 \times 2 \times 2 \times 2 = 80$ parameter combinations to explore, the average values of which are reported in Table 1.

3.2 Evaluation

In order to achieve our goal of automated identification and alignment for the purpose of reconstructing tune family evolution, we need a method of quantifying how well a given alignment captures the manual judgments of experts. The goal is to maximize both the degree of match in the alignment within tune families and the degree of accuracy in separating between tune families.

3.2.1 Sequence alignment

To evaluate alignment within tune families, we need a measure of the degree to which the similarities between sequences captured by the automated alignment matched similarities captured by the manual alignments. For this, we adopted the Mantel distance matrix correlation test [13]. The Mantel r -value is identical to a standard Pearson correlation r -value, but the Mantel significance test controls for the fact that pairwise distance values in a distance matrix are not independent of one another.

We adopted the simplest method for comparing pairs of sequences, which is by calculating their percent identity (PID). This is calculated based on the number of aligned pitches that are identical (ID) divided by the sequence length (L) according to the following equation:

$$PID = 100 \left(\frac{ID}{L_1 + L_2} \right) \quad (1)$$

This equation uses the average length of both sequences as the denominator, as this appears to be the most consistent measure of percent identity when dealing with cases where the sequences have unequal lengths due to the insertion/deletion of large segments [15] (as occurs in our dataset).

3.2.2 Tune family identification

To evaluate separation between tune families, we need a measure of the degree to which our automated clustering into tune families matches the manual tune family classifications. This needs to take into account both true positives (tunes correctly grouped into a given tune family) and false positives (tunes incorrectly grouped into a given tune family).

A method used previously by van Kranenburg et al. [25], used the true positive rate (*tpr*) and false positive rate (*fpr*) to calculate a score *J* as follows:

$$J = \frac{tpr}{1 + fpr} \quad (2)$$

Because van Kranenburg et al. did not have a method for automatically identifying boundaries between tune families, they used a “nearest neighbor” criterion to define true positives. Thus, *J* represents the proportion of tunes whose nearest neighbor (tune with highest automatically measured similarity) is also in the same (manually identified) tune family. Here we calculate this *J* score, as well as a second *J* score that more directly tests our goal of identifying boundaries between tune families.

For this second *J* score, the criterion used to define true positives is of significant sequence similarity for each pair of tunes. Significance is assessed by a random permutation test, in which the PID value for a given pair of sequence is compared against the distribution of 100 random PID values given the same sequence lengths and compositions, as calculated by randomly reordering one of the sequences [8]. Thus, when calculating this second *J* score, bold values within the boxes in Table 2 (i.e., significant sequence similarity between pairs of tunes manually identified as belonging to the same tune family) are counted as true positives, while bold values outside of the boxes (i.e., significant sequence similarity between pairs of tunes not manually identified as belonging to the same tune family) are counted as false positives.

4. RESULTS

The average scores under the different alignment parameters are shown in Table 1, with the best-performing parameter values highlighted in bold.

4.1 Sequence alignment (within-family)

The degree to which similarities within tune families captured by the automated alignment match those captured by the manual alignments of experts are indexed by the Mantel correlation *r*-values, reported in Table 1. On average, all of the alignment parameter combinations gave similarly strong correlations ranging from *r*=.82-.85.

Automated alignment parameter	Parameter value	Within-family	Between-family	
		<i>r</i>	<i>J</i> (nearest neighbor)	<i>J</i> (significance)
GOP	8	0.850	0.875	0.408
	4	0.843	0.870	0.421
	8	0.823	0.849	0.479
	12	0.833	0.877	0.497
	16	0.829	0.844	0.474
GOP:GEP ratio	2	0.834	0.862	0.462
	4	0.837	0.864	0.450
Mode	Included	0.839	0.841	0.445
	Ignored	0.832	0.885	0.467
Rhythmically unaccented notes	Included	0.841	0.964	0.587
	Ignored	0.830	0.762	0.325
Non-text notes	Included	0.838	0.873	0.460
	Ignored	0.833	0.853	0.452

Table 1. Mean values comparing different automated alignment parameters against manual ground-truth alignments. Best-performing values are highlighted in bold. See Methods for details.

4.2 Tune family identification (between-family)

The degree to which the automated algorithms were able to separate between tune families is indexed by the *J* scores, reported in the right-hand columns of Table 1. Using gap opening penalties of 12, ignoring mode, including non-text notes, and especially including rhythmically unaccented notes all improved tune-family identification. GOP:GEP ratios of 4 gave slightly higher *J* scores using the nearest neighbor criterion, but a ratio of 2 gave higher *J* scores using the more crucial criterion of significant pairwise sequence similarity. The specific parameter combination combining the best-performing parameter values - GOP=12, GOP:GEP ratio=2, ignoring mode, including rhythmically unaccented notes and including non-text notes - resulted in a Mantel correlation of *r*=.83 and *J* scores of *J*=1 and *J*=.64 for the nearest neighbor and significance criteria, respectively.

It was not possible to directly compare all parameters using the approach presented in [25], in part because the approach in [25] is based on sequences of pairwise melodic intervals, whereas the manual alignments that formed our ground-truth dataset were based on sequences of individual notes in relation to the tonic (i.e., tonic intervals). However, it was possible to directly compare between-family identification *J* scores using the best-performing parameter combination listed above, but using sequences of melodic intervals rather than tonic intervals. This melodic interval approach resulted in *J* scores of *J*=.88 and *J*=.33 for the nearest neighbor and significance criteria, respectively. These values were somewhat lower than the respective values using our tonic interval approach (*J*=1 and *J*=.64). However, further analyses are required to determine the degree to which incorporating

	1A	1B	1C	1D	1E	1F	2A	2B	2C	2D	2E	2F	2G	2H	2I	2J	2K	3A	3B	3C	3D	4A	4B	4C	4D	4E			
1A		33	45	42	52	38																							
1B	51		29	37	31	28																							
1C	59	47		34	28	38																							
1D	47	47	48		40	32																							
1E	62	54	43	45		48																							
1F	53	43	50	48	61																								
2A	41	36	34	37	34	36		49	32	23	27	19	19	18	13	15	16												
2B	35	38	44	37	45	38	54		51	50	31	25	23	26	20	18	21												
2C	40	49	41	39	40	41	47	57		44	41	23	34	28	28	21	16												
2D	33	41	42	34	33	35	45	54	61		29	19	19	26	22	18	12												
2E	31	34	43	39	36	42	45	48	57	44		32	27	21	22	21	23												
2F	43	37	41	36	46	34	39	48	41	45	35		28	16	22	22	29												
2G	38	34	41	34	39	31	34	36	42	40	41	55		34	33	43	29												
2H	31	33	30	31	38	30	37	45	41	43	33	36	47		36	62	37												
2I	44	35	34	34	45	36	28	28	42	35	39	30	35	46		44	24												
2J	40	38	28	29	38	35	26	35	39	34	31	39	55	62	49		41												
2K	36	34	35	30	28	31	31	41	46	45	34	43	45	48	30	39													
3A	32	51	36	37	29	33	31	40	42	34	35	35	42	38	31	38	43		64	44	47								
3B	40	40	35	36	32	30	36	43	46	40	38	39	40	36	33	41	32	61		57	55								
3C	42	42	38	35	40	45	30	38	34	33	38	41	39	25	29	35	39	51	62		73								
3D	38	45	37	44	37	38	25	36	30	43	37	44	29	28	23	36	31	56	60	67									
4A	40	40	28	31	39	40	26	29	34	27	31	28	27	31	40	32	27	27	29	29	23			32	39	35	33		
4B	32	29	33	38	39	36	27	29	35	28	39	30	27	24	30	30	22	35	32	28	38	40			43	45	44		
4C	31	23	36	33	31	40	26	31	28	27	38	34	18	24	21	19	25	23	29	31	30	37	52			67	61		
4D	35	26	27	30	33	36	28	26	36	28	35	31	26	25	27	22	21	26	30	21	24	41	55	65			78		
4E	32	32	35	28	39	32	27	30	40	32	41	33	26	27	32	29	23	31	33	36	28	42	62	56	62				

Table 2. Pairwise percent identity scores among the 26 tunes. Tunes are labeled based on manual classifications by musicologists [3], [12]. Numbers correspond to the four tune families (1=“Brave Donnelly”, 2=“Job of Journeywork”, 3=“Oiwake”, 4=“Okesa”), letters correspond to the different variant tunes within each family. The values in the lower triangle are based on automated alignments using the best-performing parameters (GOP=12, GOP:GEP ratio=2, ignoring mode, including rhythmically unaccented notes and including non-text notes). The values in the upper triangle are based on manual alignments. Inter-tune family manual values are not shown because manual alignments were only done within tune families. Solid borders indicate automatically identified tune families in which at least three tunes are all significantly similar to one another. When these did not capture all tunes in a manually identified tune family, the manually identified boundaries are shown using dashed borders. Bold values indicates pairs whose similarities are significant at $P < .05$.

more fine-grained weighting of intervals, rhythmic information, etc. of the type used in [25] affects tune family identification using both melodic interval and tonic interval approaches.

4.3 Overall reconstruction of tune family evolution

The results of the top-performing parameter combination listed above are compared against manual classifications

in Table 2 and Fig. 3. The lower triangle in Table 2 gives the raw pairwise sequence identity values, using bold text to indicate pairs of sequences whose similarities were statistically significant, while the upper diagonal gives within-family sequence identity values for the manual alignments. The mean percent identity values were somewhat higher for the automated alignments than the manual alignments within each family (45.7% vs. 33.7%, respectively). This presumably reflects the automated alignment identifying more false links, although in some cases it may also be identifying better alignments than the manual ones. Comparison with manual alignments conducted by different musicologists may help to clarify this issue in the future.

Fig. 3 summarizes the information in Table 2 visually using a NeighborNet diagram. NeighborNet is a type of phylogenetic network that is similar to a neighbor-joining tree, but allows visualization of conflicting non-tree like structure (“reticulation”). 100% of the tunes (26/26) were correctly grouped such that their nearest neighbor was a member of the same tune family, and the sub-grouping of tune family 2 also corresponded to Bayard’s sub-grouping into a “long” and “short” version. However, only 85% (22/26) of these tunes were automatically grouped into a tune family using the criterion that all pairs within a family must be significantly similar to one another. Using this criterion also mis-identified the “long” and “short” versions of tune family 2 as two distinct tune

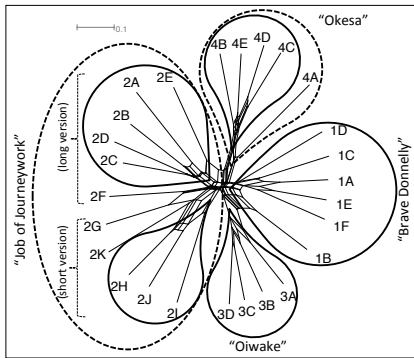


Figure 3. A NeighborNet visualization of the phylogenetic relationships among the 26 tunes automatically identified by the best-performing alignment algorithm. See Table 2 for explanation of tune labels 1A-4E and solid/dashed lines.

families. Joining families into “superfamilies” when only one or a few members have significant similarities to members of other families [8] would join the “long” and “short” versions into a superfamily, but would also join all the tune families into this superfamily.

5. DISCUSSION AND FUTURE WORK

Although previous research suggested that tune family identification was “too ambitious to perform automatically” [24], we have presented an automated approach that successfully recovers most of the key relationships within and between tune families identified manually by musicologists. Our approach adapts sequence alignment algorithms for protein family identification to successfully delineate the boundaries separating groups of melodies that share similar sequences of pitches due to descent from a common ancestor.

Our approach correctly identified three out of the four manually identified tune families, as well as both the “long version” and “short version” sub-groups of the fourth “Job of Journeywork” tune family. However, our automated approach failed to unite these sub-groups into a single tune family, instead splitting them into two tune families. The “Job of Journeywork” tune family was specifically chosen by Bayard [3] to present one of the most complicated examples of tune family evolution, including several measures that were deleted from the beginning of the “long version” and added to the end of the “short version”. Hence, this type of complex evolution may require more complex algorithms and/or the incorporation of expert knowledge beyond the basic pitch sequence information encoded in the simplified model used here. However, the fact that our approach captured the relationships among the four tunes from the “Oiwake” tune family, despite the fact that this family contained both internal and

terminal insertion/deletion events of substantial length, suggests that our approach is still able to capture fairly complicated patterns of musical evolution.

One area for improvement of our method is that the false positive rate is somewhat high (see Table 2). We believe that this may be due to the fact that our method is designed primarily to distinguish between chance and common ancestry, and does not do a very good job of distinguishing between common ancestry and convergent evolution. Hence, it appears likely that many of the false positives are due to stylistic similarities shared between unrelated tunes that share similar scales and motivic patterns (e.g., 1A and 2A, both Irish tunes in a diatonic major scale). Horizontal transmission and/or convergent evolution of such traits among phylogenetically unrelated groups have long been known to complicate analysis of tune family evolution [3], [7]. Horizontal transmission and convergent evolution are challenges shared with language evolution and genetic evolution, and may benefit from methods developed in these fields [1].

In the future we hope to extend our approach to larger datasets, and to incorporate more-sophisticated models of cultural evolution and sequence alignment [1], more-nuanced weighting of musical information (e.g., beyond simple match/mismatch models of pitch, rhythm, and text [24-26]), and higher-level units of musical structure and meaning. In music, as in genetics, the individual notes that make up the sequences have little meaning in themselves. The phylogenetic analysis of sequences is thus merely the starting point from which to understand how and why these sequences combine to form higher-level functional units (e.g., motives, phrases) that co-evolve with their song texts and cultural contexts of music-making as they are passed down from singer to singer through centuries of oral tradition. Using such information, we hope to not only identify previously unknown tune family relationships on a wide scale, but also to carefully reconstruct the histories and mechanisms of tune family evolution to identify general processes governing the cultural evolution of music. The general nature of our approach means that it should be applicable not only to folk music, but also to art music (e.g., European classical music [28], Japanese *gagaku* [14]) and popular music (e.g., copyright disputes [20]). Understanding the cultural evolution of music should help to identify the mechanisms that govern stability and creativity of aesthetic forms, as well as to use this knowledge to help musicians and musical cultures struggling to adapt their intangible cultural heritage to today’s globalized world.

Acknowledgments: We thank H. Oota and H. Matsumae for advice on adapting genetic sequence alignment algorithms to music, and S. Brown, T. Currie, and four anonymous reviewers for comments on previous drafts of this paper. Funding support for this work was provided by a Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT) scholarship to P.E.S and a Rutherford Discovery Fellowship to Q.D.A.

6. REFERENCES

- [1] Q. D. Atkinson and R. D. Gray, "Curious parallels and curious connections: Phylogenetic thinking in biology and historical linguistics." *Systematic Biology*, vol. 54, no. 4, pp. 513–526, 2005.
- [2] S. P. Bayard, "Prolegomena to a study of the principal melodic families of British-American folk song." *Journal of American Folklore*, vol. 63, no. 247, pp. 1–44, 1950.
- [3] S. P. Bayard, "Two representative tune families of British tradition," *Midwest Folklore*, vol. 4, no. 1, pp. 13–33, 1954.
- [4] C. Boiles, "Reconstruction of proto-melody," *Anuario Interamericano de Investigacion Musical*, vol. 9, pp. 45–63, 1973.
- [5] B. H. Bronson, *The traditional tunes of the Child ballads: With their texts, according to the extant records of Great Britain and America [4 volumes]*. Princeton, NJ: Princeton University Press, 1959-1972.
- [6] B. H. Bronson, "Toward the comparative analysis of British-American folk tunes," *Journal of American Folklore*, vol. 72, no. 284, pp. 165–191, 1959.
- [7] J. R. Cowdery, "A fresh look at the concept of tune family," *Ethnomusicology*, vol. 28, no. 3, pp. 495–504, 1984.
- [8] R. F. Doolittle, "Similar amino acid sequences: Chance or common ancestry?," *Science*, vol. 214, no. 4517, pp. 149–159, 1981.
- [9] P. Ferraro and P. Hanna, "Optimizations of local edition for evaluating similarity between monophonic musical sequences," *Proceedings of the International Conference on Computer-Assisted Information Retrieval*, pp. 64–69, 2007.
- [10] International Folk Music Council, "Resolutions: Definition of folk music," *Journal of the International Folk Music Council*, vol. 7, p. 23, 1955.
- [11] A. Kaneshiro, "*Kashi onretsubou ni yoru Oiwakebushi no hikaku* [Comparison of Oiwake melodies through lyric-note alignment]," *Minzoku Ongaku*, vol. 5, no. 1, pp. 30–36, 1990.
- [12] K. Machida and T. Takeuchi, Eds., *Esashi Oiwake to Sado Okesa: Min'yo genryuukou [Folk song geneologies: Esashi Oiwake and Sado Okesa] [4 LPs]*. Kawasaki: Columbia. AL-5047/50, 1965.
- [13] N. Mantel, "The detection of disease clustering and a generalized regression approach," *Cancer Research*, vol. 27, no. 2, pp. 209–220, 1967.
- [14] A. Marett, "Togaku: Where have the Tang melodies gone, and where have the new melodies come from?," *Ethnomusicology*, vol. 29, no. 3, pp. 409–431, 1985.
- [15] A. C. W. May, "Percent sequence identity: The need to be explicit," *Structure*, vol. 12, pp. 737–738, May 2004.
- [16] M. Mongeau and D. Sankoff, "Comparison of musical sequences," *Computers and the Humanities*, vol. 24, pp. 161–175, 1990.
- [17] S. B. Needleman and C. D. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of Molecular Biology*, vol. 48, pp. 443–453, 1970.
- [18] NHK (Nippon Hōsō Kyōkai), Ed., *Nihon min'yō taikan [Japanese folk song anthology] [13 volumes]*. Tokyo: NHK, 1944-1994.
- [19] R Development Core Team, *R: A language and environment for statistical computing*. Vienna: R Foundation for Statistical Computing, 2011.
- [20] M. Robine, P. Hanna, P. Ferraro, and J. Allali, "Adaptation of string matching algorithms for identification of near-duplicate music documents," *Proceedings of the Workshop on Plagiarism Analysis, Authorship Identification, and Near-Duplicate Detection*, pp. 37–43, 2007.
- [21] P. E. Savage and S. Brown, "Toward a new comparative musicology," *Analytical Approaches to World Music*, vol. 2, no. 2, pp. 148–197, 2013.
- [22] J. D. Thompson, D. G. Higgins, and T. J. Gibson, "CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice," *Nucleic Acids Research*, vol. 22, no. 22, pp. 4673–4680, 1994.
- [23] J. Urbano, J. Lloréns, J. Morato, and S. Sánchezcuadrado, "Melodic similarity through shape similarity," *Proceedings of the International Symposium on Computer Music Modeling and Retrieval*, pp. 338–355, 2011.
- [24] P. van Kranenburg, J. Garbers, A. Volk, F. Wiering, L. Grijp, and R. C. Veltkamp, "Towards integration of MIR and folk song research," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 505–508, 2007.
- [25] P. van Kranenburg, A. Volk, and F. Wiering, "A comparison between global and local features for computational classification of folk song melodies," *Journal of New Music Research*, vol. 42, no. 1, pp. 1–18, 2013.
- [26] P. van Kranenburg, A. Volk, F. Wiering, and R. C. Veltkamp, "Musical models for folk-song melody alignment," *Proceedings of the International Symposium on Music Information Retrieval*, pp. 507–512, 2009.
- [27] A. Whiten, R. A. Hinde, C. B. Stringer, and K. N. Laland, *Culture evolves*. Oxford: Oxford University Press, 2012.
- [28] H. F. Windram, T. Charlston, and C. J. Howe, "A phylogenetic analysis of Orlando Gibbons's Prelude in G," *Early Music*, vol. 42, no. 4, pp. 515–528, 2014.

EVALUATION OF ALBUM EFFECT FOR FEATURE SELECTION IN MUSIC GENRE RECOGNITION

Igor Vatolkin

TU Dortmund

Department of Computer Science

igor.vatolkin@udo.edu

Günter Rudolph

TU Dortmund

Department of Computer Science

guenter.rudolph@udo.edu

Claus Weihs

TU Dortmund

Faculty of Statistics

claus.weihs@udo.edu

ABSTRACT

With an increasing number of available music characteristics, feature selection becomes more important for various categorisation tasks, helping to identify relevant features and remove irrelevant and redundant ones. Another advantage is the decrease of runtime and storage demands. However, sometimes feature selection may lead to “over-optimisation” when data in the optimisation set is too different from data in the independent validation set. In this paper, we extend our previous work on feature selection for music genre recognition and focus on so-called “album effect” meaning that optimised classification models may overemphasize relevant characteristics of particular artists and albums rather than learning relevant properties of genres. For that case we examine the performance of classification models on two validation sets after the optimisation with feature selection: the first set with tracks not used for training and feature selection but randomly selected from the same albums, and the second set with tracks selected from other albums. As it can be expected, the classification performance on the second set decreases. Nevertheless, in almost all cases the feature selection remains beneficial compared to complete feature sets and a baseline using MFCCs, if applied for an ensemble of classifiers, proving robust generalisation performance.

1. INTRODUCTION

Among many different scenarios for automatic classification of music data (we refer to [4] for an introduction to content-based music information retrieval and an overview of related tasks), the recognition of high-level music categories such as music genres and styles is one of the most prominent and user-related applications. Probably the first study on automatic categorisation of music was addressed to distinguish between several classical and popular pieces [22]. After the seminal work of Tzanetakis and Cook on classifying musical data into a hierarchy of 25 music genres and speech categories [38] many efforts were spent to

enhance the methods, develop new features, and integrate actual techniques from machine learning research [42]. [37] lists several hundreds of studies related only to the recognition of genres. Since 2005, audio genre classification belongs to tasks of the annual MIREX contest [6].

The operating principle of supervised classification is based on two stages: the training of a classification model \mathcal{CT} and its application \mathcal{C} on uncategorised data:

$$\begin{aligned}\mathcal{CT} : (\mathbf{X} \in \mathbb{R}^{F \times T_{TR}}, \mathbf{y}_L \in \mathbb{R}^{T_{TR}}) &\mapsto \mathcal{M}, \\ \mathcal{C} : (\mathbf{X} \in \mathbb{R}^{F \times T}, \mathcal{M}) &\mapsto \mathbf{y}_P \in \mathbb{R}^T.\end{aligned}\quad (1)$$

Given a set of F numeric data characteristics, or features, for T_{TR} data instances (also referred to as classification windows) resulting in the feature matrix \mathbf{X} , and the corresponding labels \mathbf{y}_L , the training stage identifies relevant dependencies between features and labels and stores them as a model \mathcal{M} . Some approaches are based on the estimation of probability-based distribution of features (Naive Bayes) or boundaries between data instances of different categories (support vector machines); for an overview of classification approaches see, e.g., [13, 43]. Once the classification models are saved, they can be applied to classify T unlabelled data instances represented by the same F previously extracted features.

Music classification can be carried out using features from different sources. For instance, the score allows a precise estimation of harmonic, instrumental, and rhythmic descriptors of music pieces, but it is not always available for popular music. Meta data, cultural features, or tags provide another source of information, but are sometimes incomplete or erroneous. Audio features can be extracted for every digitised music piece, and many classification approaches are limited to or focused on this kind of features [9, 18, 19, 21, 27, 34, 36, 38, 40]. Another advantage of these characteristics is that they are not dependent on the popularity of a song, availability of the score, or Internet connection for the download of metadata. Even if audio features typically require high computing efforts for their extraction, these costs can be reduced to a certain degree if the extraction is done offline or on a server farm. In that case only the time for the training and the application of classification models will influence a user’s satisfaction during the definition of new categorisation tasks. For these reasons we have limited the scope of this study to audio features only.



© Igor Vatolkin, Günter Rudolph, Claus Weihs.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Igor Vatolkin, Günter Rudolph, Claus Weihs. “Evaluation of Album Effect for Feature Selection in Music Genre Recognition”, 16th International Society for Music Information Retrieval Conference, 2015.

Having a large number of available descriptors at hand, individual features may be very important or completely useless depending on a categorisation task. As the combination of features from different sources may increase the classification quality (e.g., as shown for audio, symbolic, and cultural features in [26]), the inclusion of features from many sources would lead to an increased number of both relevant and irrelevant features. If the number of irrelevant features would become too high, the classification quality may suffer because the probability increases that some irrelevant features are identified as relevant by chance [13, 43]. A solution is to start with a sufficiently large initial feature set and to remove irrelevant and noisy characteristics for a current category by means of feature selection (FS). Other benefits of FS are that classification models created with less features often require less storage space, the classification is done faster, and the danger of overfitting towards the training set may be reduced using a proper evaluation of models and feature sets [3].

In our previous work we have applied feature selection for the recognition of music genres and styles and measured a significant increase of classification performance compared to complete feature sets [39]. For the final evaluation of models optimised with feature selection, we used an independent validation set with tracks not used for model training and feature selection. The motivation for an independent evaluation in music classification is discussed in [9]. However, strictly observed, the validation set used in our previous experiments was not completely independent: due to the limited size of our music database, music pieces for validation were different from training and optimisation sets, but randomly selected from the same albums. Therefore, a danger existed that optimised classification models would have an especially high performance on music pieces of the same artists and albums.

Such effect was observed in [30] for the recognition of genres. Also the tags of songs belonging to the same albums may have higher co-occurrences as inspected in [20]. Further investigations showed interesting results on the difference between album and artist effect for music databases of different sizes [10] as well as varying impact of artist filter with regard to music from different geographic locations around the world [15]. However, none of these studies explicitly evaluated the sensitivity of FS to artist/album effect using a large number of features. Such evaluations can be promising in future, in particular because both latter papers stated differences in measured artist effect for different feature groups, even if the overall numbers of integrated features were not very high.

Thus, the idea behind this study was to re-evaluate the measured advantage of feature selection using a new “album-independent” validation set and to estimate the album effect for different music categories. In the next section, we outline basic concepts of feature selection and refer to several applications. Section 3 describes the setup of the study. In Section 4, the results and the album effect on feature selection are discussed. We conclude with a brief summary of the work and outline steps for future research.

2. FEATURE SELECTION

For an exhaustive introduction into feature selection methods see [12]. In general, the task of feature selection is to find an optimal feature subset indicated by the binary vector \mathbf{q} ($q_i = 1$ for the i -th feature to be selected, otherwise $q_i = 0$), so that some relevance function, or evaluation criterion m (e.g., classification error) is minimised. The functions to maximise (e.g., accuracy) can be easily adapted for minimisation. We define the task of feature selection as:

$$\mathbf{q}^* = \arg \min_{\mathbf{q}} [m(\mathbf{y}_L, \mathbf{y}_P, \Phi(\mathbf{x}, \mathbf{q}))], \quad (2)$$

where $\Phi(\mathbf{x}, \mathbf{q})$ corresponds to the subset of the original feature vector \mathbf{x} . $\mathbf{y}_L \in [0; 1]$ are the labelled category relationships of classification instances, and $\mathbf{y}_P \in [0; 1]$ are the predicted category relationships. Note that in general m may not necessarily depend on labels, e.g., if the correlation between features is used as selection criterion, or if labels are not available (as in unsupervised classification).

Feature selection with regard to only one evaluation criterion may lead to a decrease of performance for other ones. For example, classification models built with too many features may have smaller classification errors for a specific data set, but be slower and have a poor generalisation performance on other data. Therefore, several relevance functions or objectives m_1, \dots, m_O may be considered for simultaneous optimisation:

$$\mathbf{q}^* = \arg \min_{\mathbf{q}} [m_1(\mathbf{y}_L, \mathbf{y}_P, \Phi(\mathbf{x}, \mathbf{q})), \dots, m_O(\mathbf{y}_L, \mathbf{y}_P, \Phi(\mathbf{x}, \mathbf{q}))]. \quad (3)$$

In literature, individual features are often referred to as relevant or redundant w.r.t. the performance of a Bayesian classifier which predicts labels based on a probabilistic distribution of feature vectors. For a given feature set X , a feature subset $X' \subset X$ is called *relevant*, iff its removal will decrease the performance of a Bayesian classifier:

$$\begin{aligned} P(y_P | y_L = y_P, X) &< P(y_P | y_L = y_P, X \setminus X') \text{ and} \\ P(y_P | y_L \neq y_P, X) &> P(y_P | y_L \neq y_P, X \setminus X'). \end{aligned} \quad (4)$$

A *redundant* feature subset X' can be replaced without decrease of a Bayesian classifier’s performance by at least one subset S , which does not contain X' :

$$\exists S \subseteq X, X' \cap S = \emptyset : P(y_P | X) = P(y_P | S). \quad (5)$$

The equations (4)–(5) can be adapted to any relevance function, describing a decrease of performance after the removal of relevant features and retaining it after the removal of redundant features.

FS is a very complex task: for F features, the number of all possible non-empty feature subsets is $2^F - 1$, and the related problems were described as NP-hard [1, 14]. Therefore, metaheuristics like evolutionary algorithms (EA) [33] which simulate the natural evolution based on principles of recombination (keeping the positive characteristics of solutions) and mutation (exploring the search space using

some random procedure) are a possible remedy. EAs have proven their ability to solve many complex optimisation tasks, among others for data mining and classification [28]. The first application of EAs for FS was introduced in [35], and EAs were recommended for sets with more than 100 features in [16] after the comparison of 18 FS methods.

FS has been already often applied for music classification, for example for the recognition of musical instruments (44 features) [5], moods (66 features) [34], or several classification tasks (between 60 and 1140 features) [25]. Evolutionary FS was integrated in music classification for the first time in [11] and was applied also in later studies, e.g., [8, 27, 36]. The first application of evolutionary multi-objective algorithms to FS for the simultaneous minimisation of the number of features and misclassification rate was proposed in [7]. In music classification, multi-objective evolutionary feature selection was introduced in [40] for genre categorisation and later for the recognition of instruments [41].

In the following, we will describe the study to measure the impact of two-objective FS (minimisation of the classification error and the number of features) on the classification into music genres and styles. Classification results are compared to models built with full feature sets and a baseline with MFCCs. Further, we will investigate the sensitivity of the proposed method to the album effect.

3. EXPERIMENTAL SETUP

3.1 Categorisation Tasks

We distinguish between music genres and styles provided by AllMusicGuide, where a track may belong to one music genre and up to several music styles which are more specific and are typically harder to predict.

Our main database for experiments consists of 120 albums with approximately one third of commercial popular music (45 Pop/Rock albums) as well as tracks of several other genres for a better evaluation of generalisation performance (15 albums of each genre Classic, Electronic, Jazz, Rap, and R&B). For the evaluation of the album effect the database was extended with 120 songs from albums of other artists but the same genre and similar style distribution. It is important to mention that we use our own database, because many publicly available ones were not well suited for this work. Several databases contain only segments of songs so that it is not possible to extract features from long frames (e.g., structural complexity, see the next section). Others are strongly biased towards certain genres or are expensive because of a large share of commercial music. These problems could be in principle avoided using data sets with features only (e.g., Echo Nest descriptors). However, a sufficiently large number of audio features is necessary to measure the impact of feature selection, and many descriptors are developed by ourselves being not available in freely distributed feature sets.

We distinguish between training, optimisation, and two test sets (all of them are disjoint on track level, i.e. it is not permitted to have the same track in more than one

set). Each classification model is *trained* from 20 tracks, 10 of which belong to the category to predict (positive examples), and 10 do not belong to it (negative examples). These small training sets are motivated by the real-world situation, where a listener would like to omit high efforts for the labelling of ground truth. On the other side, music pieces have strong variations on different levels (instrumentation, vocal segments, harmony, etc.) and we build classification instances from music intervals of 4 s with 2 s overlap, so that 20 tracks contribute to more than 2,000 classification instances. The data set for the identification of relevant features is the *optimisation* set of 120 songs, each of them selected randomly from the 120 albums. The final evaluation of feature sets after feature selection is done either on 120 *test* tracks randomly selected from the original albums (test set TS) or 120 tracks from other artists (test set TSAI). Thus, the overall number of tracks for each classification experiment was equal to 260. The exact lists of tracks are available on our web site¹.

3.2 Features

Two large audio feature sets are used as baselines to compare them with sets optimised by means of feature selection. For exact definitions and references please see [39]. The third baseline set is built with MFCCs which are often used for music classification [18].

The first large set comprises low-level audio signal descriptors. Such features can be roughly grouped into timbre, rhythmic, and pitch characteristics [38]. We extend this categorisation to ‘timbre and energy’, ‘chroma and harmony’, ‘temporal and correlation characteristics’, and ‘rhythm’. Table 1 provides examples of features for different extraction domains and lists numbers of corresponding feature dimensions. Because we estimate the mean and the standard deviation of each feature vector in a classification window, the original number of 318 dimensions leads to 636 features used for the training of categorisation models.

The second set contains semantic audio features which are closely related to music theory and are listed in Table 2. They can be assigned to four main groups according to their properties and the extraction procedure. The first group consists of chroma-related, harmony, and chord characteristics. The second one comprises temporal, rhythmic, and structural characteristics. The third group (instruments, moods, and various high-level characteristics) relates to features estimated with supervised classification models previously optimised as described in [39]. The last group was extracted using the concept of structural complexity [24]. Here, selected interpretable musical characteristics (instrumentation, harmonic properties, etc.) are represented by a vector of base features, and estimated statistics describe the temporal progress of these vectors over large texture frames.

¹ https://ls11-www.cs.uni-dortmund.de/rudolph/mi#music_test_database

Table 1. Low-level audio features

Groups and examples of features	No.
TIMBRE AND ENERGY - TIME DOMAIN	
Linear prediction coefficients, low energy, peak characteristics	17
TIMBRE AND ENERGY - SPECTRAL DOMAIN	
Various spectral characteristics (bandwidth, centroid, etc.), tritrimulus, sub-band energy ratio	29
TIMBRE AND ENERGY - CEPSTRAL DOMAIN	
MFCCs, delta MFCCs, CMRARE modulation features [21]	101
TIMBRE AND ENERGY - PHASE DOMAIN	
Angles and distances [27]	2
TIMBRE AND ENERGY - ERB AND BARK DOMAINS	
Bark scale magnitudes, charact. of ERB bands [17]	53
CHROMA AND HARMONY	
Charact. of spectral peaks, fundamental frequency, chroma, chroma DCT-reduced log pitch (CRP) [29]	101
TEMPORAL AND CORRELATION CHARACTERISTICS	
Characteristics of periodicity peaks	3
RHYTHM	
Characteristics of fluctuation patterns [17]	12

Table 2. Semantic audio features

Groups and examples of features	No.
CHROMA AND HARMONY	
Consonance [23], tonal centroid [17], strengths of major and minor keys [17]	129
CHORD STATISTICS	
Number of different chords and chord changes in 10 s, shares of the most frequent chords [39]	5
TEMPO, RHYTHM AND STRUCTURE	
Duration of music piece, estimated number of beat, tatum, and onset events per minute, tempo, segmentation characteristics after [31]	9
INSTRUMENTS	
Identification of guitar, piano, wind, and strings [41]	32
MOODS	
Aggressive, confident, energetic, etc. [39]	64
VARIOUS HIGH-LEVEL CHARACTERISTICS	
Singing characteristics, effects distortion, characteristics of melodic range [32]	128
STRUCTURAL COMPLEXITY	
Chord, harmony, instruments, tempo and rhythm complexity [39]	70

3.3 Algorithms and Evaluation

The exhaustive tuning of classification methods was beyond the scope of this study - however it was important to test the impact of feature selection and the album effect using classifiers with different operating methods. After preliminary studies, we selected four algorithms. Decision tree C4.5 provides interpretable models and already includes internal feature pruning, but is rather slow. Random forest (RF) creates a large number of unpruned trees based on a randomly drawn subset of features. It is often superior to C4.5 w.r.t. classification quality and is faster, but classification models are not the same if trained another time and are not interpretable. Naive Bayes (NB) is very fast and leads to comprehensible models, especially if they are created from interpretable semantic features. On the other side, it is a probabilistic method which treats feature distributions independently from each other, and clas-

sification performance is usually lower. Finally, support vector machine (SVM) is in many cases the state-of-the-art method, which achieves the best classification results. However, for the best performance it requires parameter tuning, is slower than other methods, and models have a lower interpretability.

The following two criteria are minimised during feature selection. Because of imbalanced distribution of songs in the optimisation and test sets, the balanced relative error m_{BRE} measures classification quality:

$$m_{BRE} = \frac{1}{2} \left(\frac{FN}{TP + FN} + \frac{FP}{TN + FP} \right), \quad (6)$$

where TP is a number of true positives (tracks belonging to a category and predicted as belonging to it), TN is a number of true negatives (tracks not belonging to a category and predicted as not belonging to it), FP is a number of false positives (tracks not belonging to a category and predicted as belonging to it), and FN is a number of false negatives (tracks belonging to a category and predicted as not belonging to it).

The predicted relationships of tracks to categories are estimated by major voting across all corresponding classification windows:

$$y_P(\mathbf{x}_1, \dots, \mathbf{x}_{T_S}; j) = \left[\frac{\sum_{i=1}^{T_S} y_P(\mathbf{x}_i) - 0.5}{T_S} \right], \quad (7)$$

where T_S is the number of classification instances in the song j and \mathbf{x}_i describes the feature vector of instance i .

The second optimisation criterion is the selected feature rate m_{SFR} :

$$m_{SFR} = \frac{|\Phi(\mathbf{x}, \mathbf{q})|}{|X|}, \quad (8)$$

where $|\Phi(\mathbf{x}, \mathbf{q})|$ is the number of selected features and $|X|$ the number of all features. m_{SFR} is a rough estimator for runtime and storage demands (classification using a model with more features is typically slower), but may also correlate with the generalisation performance of classification models: models built with less features have a lower tendency to be overfitted towards the training set if the optimisation of feature selection is done using an independent song set.

The feature selection method itself is based on a multi-objective evolutionary algorithm SMS-EMOA [2]. The output is the set of non-comparable feature subsets: the first with the largest m_{SFR} and smallest m_{BRE} , and the last with the smallest m_{SFR} and largest m_{BRE} ². Because we focus here on the measurement of album effect having regard to classification error, the discussion of results in the next section is based on subsets with the smallest m_{BRE} . These subsets contain smallest errors achieved for as small feature subsets as possible.

² As we minimise both m_{SFR} and m_{BRE} , an example of two non-comparable (also referred to as *non-dominated*) subsets is, e.g., a subset with $m_{SFR} = 0.05$, $m_{BRE} = 0.20$ and another one with $m_{SFR} = 0.10$, $m_{BRE} = 0.15$. The first subset is built with less features and the second one has a smaller classification error.

Table 3. Errors of optimised feature sets and comparison to baselines (smaller values are better). For details see the text.

Categorisation tasks		Data set	LOW-LEVEL FEATURES			SEMANTIC FEATURES		
			\tilde{m}_{BRE}	Φ_{LL}	Φ_{MFCC}	\tilde{m}_{BRE}	Φ_{SEM}	Φ_{MFCC}
RECOGNITION OF GENRES								
Classic	TS	0.0127	41.91	33.07	0.0137	37.43	35.68	
	TSAI	0.0175	39.95	32.47	0.0270	57.20	50.09	
Electronic	TS	0.0928	66.19	48.91	0.1191	59.25	62.78	
	TSAI	0.1040	82.47	56.37	0.1275	56.52	69.11	
Jazz	TS	0.0497	66.89	47.56	0.0605	69.86	57.89	
	TSAI	0.1192	107.00	89.42	0.1113	49.47	83.50	
Pop	TS	0.1291	74.71	68.34	0.1270	43.94	67.23	
	TSAI	0.1599	41.20	75.53	0.1353	27.91	63.91	
Rap	TS	0.0508	70.26	56.31	0.0650	76.29	72.06	
	TSAI	0.0475	72.74	88.29	0.0579	56.54	107.62	
R&B	TS	0.1570	89.82	74.09	0.1484	76.85	69.93	
	TSAI	0.1337	84.73	56.29	0.1486	73.67	62.57	
RECOGNITION OF STYLES								
AdultContemporary	TS	0.1192	67.31	55.94	0.1344	57.02	63.07	
	TSAI	0.1906	64.83	81.45	0.1860	64.27	79.49	
AlbumRock	TS	0.0900	65.41	46.68	0.1066	51.15	55.29	
	TSAI	0.1225	61.47	49.04	0.1617	50.73	64.73	
AlternativePopRock	TS	0.1066	70.92	49.67	0.1092	54.19	50.89	
	TSAI	0.1746	71.47	81.40	0.1818	64.10	84.76	
ClubDance	TS	0.1551	82.41	74.35	0.1389	55.02	66.59	
	TSAI	0.1398	70.25	54.69	0.1465	64.65	57.32	
HeavyMetal	TS	0.0839	59.25	92.20	0.0778	56.25	85.49	
	TSAI	0.1192	59.22	85.26	0.0991	55.06	70.89	
ProgRock	TS	0.1072	64.00	47.43	0.0973	53.52	43.04	
	TSAI	0.1780	57.68	65.56	0.2039	59.85	75.10	
SoftRock	TS	0.1104	67.28	45.19	0.1197	53.13	49.00	
	TSAI	0.1752	69.91	65.28	0.1498	62.39	55.81	
Urban	TS	0.1038	76.95	74.67	0.0837	57.06	60.22	
	TSAI	0.1541	59.09	58.81	0.1553	51.87	59.27	

4. DISCUSSION OF RESULTS

4.1 Table with Results

Table 3 provides the summary of results and is organised as follows. The first column lists categorisation tasks. The second column indicates whether the album-dependent test set TS or album-independent test set TSAI was used for the final validation. Columns 3-5 describe results with low-level features. In the column 3 the “mean best” error \tilde{m}_{BRE} is listed. The mean is here calculated across 10 statistical repetitions: because evolutionary FS is based on random decisions, the results are not the same for each run. So the value of $\tilde{m}_{BRE} = 0.0127$ corresponds to the expected best m_{BRE} after the application of FS. The best means that we take into account feature subsets with the smallest m_{BRE} and the largest m_{SFR} across compromise solutions identified with a multi-objective selection approach (see the previous section).

Entries in columns 4 and 5 measure the relative reduction of \tilde{m}_{BRE} compared to complete set of low-level features, Φ_{LL} , and set of MFCCs, Φ_{MFCC} . Smaller values are better. For example, in the first line $\tilde{m}_{BRE} = 0.0127$ corresponds to 41.91% of the error of the model which uses all low-level descriptors ($m_{BRE} = 0.0303^3$). Similarly,

³ Please note that we use an ensemble of four classifiers and select the best one for each task. Using a complete feature set for the category Classic leads to $m_{BRE} = 0.0303$ if trained with random forest; for example, using naive Bayes leads to $m_{BRE} = 0.0695$, so that the error of

\tilde{m}_{BRE} is reduced to 33.07% of the error of the model built with MFCCs only.

Columns 6-8 contain values of \tilde{m}_{BRE} for models built with semantic features and the reduction of error compared to full set of semantic features Φ_{SEM} and Φ_{MFCC} .

4.2 Album Effect and Two Cases where Feature Selection Fails

As it could be expected, classification errors increase for most of categories if we switch from the test set TS to TSAI. The advantage of optimised feature subsets compared to baselines (columns 4,5,7,8) is often decreased, but not always. For instance, despite of a larger error for AdultContemporary using TSAI (0.1906 against 0.1192), the advantage of optimised low-level feature subsets compared to the model with all low-level features is slightly increased (64.83 against 67.31, smaller value is better), but not if compared to the model built with MFCCs (81.45 against 55.94).

A more important observation is that in all but two cases optimised models are better than baselines (only two entries in columns 4,5,7,8 are above 100%) which means that feature subsets after FS lead to a robust reduction of error even if finally validated on the test set from inde-

the optimised combination “feature subset and classifier” is even stronger reduced if compared to a simple application of naive Bayes together with all low-level features.

pendent artists and albums. The first exception is Jazz (value of 107.00 in the 4th column): here the full Φ_{LL} set ($m_{BRE} = 0.1114$) leads to a slightly smaller error than the optimised set ($m_{BRE} = 0.1192$). This can be explained by the choice of music: in the artist-independent validation song set the category Jazz was represented rather by European Jazz, where the training and optimisation set contained rather American Jazz⁴. Another exception relates to the error of optimised subsets with semantic features compared to MFCCs for Rap (value of 107.62, column 8). This matches well the theoretical reason that MFCCs are particularly successful for the recognition of speech. The smallest error for Rap is achieved using the optimised set with low-level features (and MFCCs belong to this set): $m_{BRE} = 0.0475$.

4.3 A Further Danger for Feature Selection (or Advantage of Ensembles)

In all but two explained situations FS led to smaller errors. However, this statement holds for classification with four methods. Using an ensemble of classifiers makes often sense, and in our previous work we have already observed that there is no “winner” for all categories [39]. To examine whether the feature selection was successful for individual combinations of a classifier and a task we compared the results to baselines by means of Wilcoxon test. If no statistical advantage against a baseline has been observed for at least one of four classifiers, the corresponding entry in Table 3 is marked with an italic font. If the baseline was even better for at least one classifier, the entry is marked with a bold font. Particularly some models with MFCCs seem to provide a better generalisation performance rather than optimised feature subsets. This happens only if test set TSAI is used for the validation. In other words, optimising feature selection with an individual classifier may lead to overfitting—but in our study this case was avoided using an ensemble of several classifiers.

4.4 A Remark on Resources

Beside possible problems for feature selection discussed above, it should not be forgotten that FS provides a strong advantage against large sets of features because it helps to reduce storage and runtime demands. The advantage of smaller feature sets is that the classification is typically faster⁵. When the time expensive feature selection may be run once for each new music category, the automatic classification based on the optimised feature set can be applied on new songs over and over again. It is hard to precisely measure the reduction of computing demands, especially for experiments on different machines. As a rough mea-

⁴ We came to this explanation after the studies were accomplished. The uniform sampling of European and American Jazz tracks for optimisation and validation sets could be a better decision, but in that case it would not be possible to exactly compare the results to [39].

⁵ As we could see, a set of MFCCs is also small and is sometimes successful, so the reduction of demands on resources is not very strong here. However, all but one values in columns 5 and 8 are below 100%, and it is probably not the best idea to build classification models with MFCCs only for all possible classification tasks (styles, tags, moods, etc.)

sure we may estimate the decrease of runtime of the last FS iteration compared to the first iteration (in each iteration, a classification model is trained and validated). As an example, the mean of runtime of the last iteration divided by runtime of the first iteration for the category Classic is 15.08 for the low-level feature set and 12.57 for the semantic set (classification with C4.5), 34.55 and 31.72 (RF), 20.88 and 8.56 (NB), and 12.68 and 10.54 (SVM).

5. CONCLUSIONS AND OUTLOOK

In this work we have examined whether the success of feature selection in music classification suffers from an “album effect”, so that the properties of albums and artists rather than of target categories like genres and styles are learned. As it could be expected, the danger of such overfitting exists, and the performance is typically reduced if the validation set is built with tracks of other artists. However, if there are enough available features at hand, and feature selection is applied using an ensemble of classifiers, in all but two cases the optimised subsets helped to build classification models not only with less features, but also with smaller classification errors compared to baselines. These two cases could be theoretically explained and do not detract the general sense of feature selection - but they underline the consequence that any significant achievements in classification domain raise and fall with the design of data sets. A very simple case observed in this study was that the classification models optimised to recognise particularly American Jazz were not best suited to recognise European Jazz. In future we plan to continue our work investigating advantages and dangers of feature selection for music classification. In particular, the application on publicly available data sets is important for a reliable comparison of results. However, this is a hard task which requires compromises, e.g., limiting the set of features only to available Echo Nest descriptors. Further optimisation of algorithm parameters (e.g., larger ensembles, various kernels for SVMs) is another promising direction.

6. REFERENCES

- [1] E. Amaldi and V. Kann. On the approximability of minimizing nonzero variables or unsatisfied relations in linear systems. *Theoretical Computer Science*, 209(1-2):237–260, 1998.
- [2] N. Beume, B. Naujoks, and M. Emmerich. SMS-EMOA: Multiobjective selection based on dominated hypervolume. *European Journal of Operational Research*, 181(3):1653–1669, 2007.
- [3] B. Bischl, O. Mersmann, H. Trautmann, and C. Weihs. Resampling methods for meta-model validation with recommendations for evolutionary computation. *Evolutionary Computation*, 20(2):249–275, 2012.
- [4] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: Current directions and future challenges. *Proceedings of the IEEE*, 96(4):668–696, 2008.
- [5] J. D. Deng, C. Simmermacher, and S. Cranefield. A study on feature analysis for musical instrument classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 38(2):429–438, 2008.

- [6] J. S. Downie, A. F. Ehmann, M. Bay, and M. C. Jones. The Music Information Retrieval Evaluation eXchange: Some observations and insights. In Z. W. Ras and A. Wiecezorkowska, editors, *Advances in Music Information Retrieval*, pages 93–115. Springer, 2010.
- [7] C. Emmanouilidis, A. Hunter, and J. MacLntyre. A multiobjective evolutionary setting for feature selection and a commonality-based crossover operator. In *Proc. IEEE Congress on Evolutionary Computation (CEC)*, volume 1, pages 309–316, 2000.
- [8] S. Essid, G. Richard, and B. David. Musical instrument recognition by pairwise classification strategies. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(4):1401–1412, 2006.
- [9] R. Fiebrink and I. Fujinaga. Feature selection pitfalls and music classification. In *Proc. 7th Int'l Conf. on Music Information Retrieval (ISMIR)*, pages 340–341, 2006.
- [10] A. Flexer and D. Schnitzer. Effects of album and artist filters in audio similarity computed for very large music databases. *Computer Music Journal*, 34(3):20–28, 2010.
- [11] I. Fujinaga. Machine recognition of timbre using steady-state tone of acoustic musical instruments. In *Proc. Int'l Computer Music Conf. (ICMC)*, pages 207–210, 1998.
- [12] I. Guyon, M. Nikravesh, S. Gunn, and L. A. Zadeh, editors. *Feature Extraction. Foundations and Applications*, volume 207 of *Studies in Fuzziness and Soft Computing*. Springer, Berlin Heidelberg, 2006.
- [13] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, New York, 2009.
- [14] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.
- [15] A. Kruspe, H. Lukashevich, and J. Abeßer. Artist filtering for non-western music classification. In *Proc. 6th Audio Mostly Conference (AM)*, pages 82–86, 2011.
- [16] M. Kudo and J. Sklansky. Comparison of algorithms that select features for pattern classifiers. *Pattern Recognition*, 33(1):25–41, 2000.
- [17] O. Lartillot. *MIRtoolbox 1.4 User's Manual*. Finnish Centre of Excellence in Interdisciplinary Music Research and Swiss Center for Affective Sciences, 2012. Online resource.
- [18] B. Logan. Mel frequency cepstral coefficients for music modeling. In *Proc. 1st Int'l Symp. on Music Information Retrieval (ISMIR)*, 2000.
- [19] M. I. Mandel and D. Ellis. Song-level features and support vector machines for music classification. In *Proc. 6th Int'l Conf. on Music Information Retrieval (ISMIR)*, pages 594–599, 2005.
- [20] M. I. Mandel, R. Pascanu, D. Eck, Y. Bengio, L. M. Aiello, R. Schifanella, and F. Menczer. Contextual tag inference. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 7(Suppl.):32, 2011.
- [21] R. Martin and A. M. Nagathil. Cepstral modulation ratio regression (CMRARE) parameters for audio signal analysis and classification. In *Proc. IEEE Int'l Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 321–324, 2009.
- [22] B. Matityaho and M. Furst. Neural network based model for classification of music type. In *Proc. 18th Convention of Electrical and Electronics Engineers in Israel*, pages 4.3.4/1–4.3.4/5, 1995.
- [23] M. Mauch and S. Dixon. Approximate note transcription for the improved identification of difficult chords. In *Proc. 11th Int'l Society for Music Information Retrieval Conf. (ISMIR)*, pages 135–140, 2010.
- [24] M. Mauch and M. Levy. Structural change on multiple time scales as a correlate of musical complexity. In *Proc. 12th Int'l Society for Music Information Retrieval Conf. (ISMIR)*, pages 489–494, 2011.
- [25] R. Mayer, A. Rauber, P. J. Ponce de León, C. Pérez-Sancho, and J. M. Iñesta. Feature selection in a cartesian ensemble of feature subspace classifiers for music categorisation. In *Proc. 3rd Int'l Workshop on Machine Learning and Music (MLL)*, pages 53–56, 2010.
- [26] C. McKay. *Automatic Music Classification with jMIR*. PhD thesis, McGill University, 2010.
- [27] I. Mierswa and K. Morik. Automatic feature extraction for classifying audio data. *Machine Learning Journal*, 58(2-3):127–149, 2005.
- [28] A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay, and C. A. Coello Coello. A survey of multiobjective evolutionary algorithms for data mining: Part I. *IEEE Transactions on Evolutionary Computation*, 18(1):4–19, 2014.
- [29] M. Müller and S. Ewert. Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features. In *Proc. 12th Int'l Conf. on Music Information Retrieval (ISMIR)*, pages 215–220, 2011.
- [30] E. Pampalk, A. Flexer, and G. Widmer. Improvements of audio-based music similarity and genre classification. In *Proc. 6th Int'l Conf. on Music Information Retrieval (ISMIR)*, pages 628–633, 2005.
- [31] J. Paulus and A. Klapuri. Music structure analysis using a probabilistic fitness measure and an integrated musicological model. In *Proc. 9th Int'l Conf. on Music Information Retrieval (ISMIR)*, pages 369–374, 2008.
- [32] G. Rötter, I. Vatolkin, and C. Wehns. Computational prediction of high-level descriptors of music personal categories. In B. Lausen, D. van den Poel, and A. Ullsch, editors, *Algorithms from and for Nature and Life*, pages 529–537. Springer, 2013.
- [33] G. Rozenberg, T. Bäck, and J. N. Kok, editors. *Handbook of Natural Computing*. Springer, Berlin Heidelberg, 2012.
- [34] P. Saari, T. Eerola, and O. Lartillot. Generalizability and simplicity as criteria in feature selection: Application to mood classification in music. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1802–1812, 2011.
- [35] W. W. Siedlecki and J. Sklansky. A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10(5):335–347, 1989.
- [36] C. N. Silla Jr., A. L. Koerich, and C. A. A. Kaestner. A feature selection approach for automatic music genre classification. *International Journal of Semantic Computing*, 3(2):183–208, 2009.
- [37] B. Sturm. A survey of evaluation in music genre recognition. In *Proc. 10th Int'l Workshop on Adaptive Multimedia Retrieval (AMR)*, 2012.
- [38] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.
- [39] I. Vatolkin. *Improving Supervised Music Classification by Means of Multi-Objective Evolutionary Feature Selection*. PhD thesis, Dep. of Computer Science, TU Dortmund, 2013.
- [40] I. Vatolkin, M. Preuß, and G. Rudolph. Multi-objective feature selection in music genre and style recognition tasks. In *Proc. 13th Annual Genetic and Evolutionary Computation Conf. (GECCO)*, pages 411–418, 2011.
- [41] I. Vatolkin, M. Preuß, G. Rudolph, M. Eichhoff, and C. Wehns. Multi-objective evolutionary feature selection for instrument recognition in polyphonic audio mixtures. *Soft Computing*, 16(12):2027–2047, 2012.
- [42] C. Wehns, U. Ligges, F. Mörchen, and D. Müllensiefen. Classification in music research. *Advances in Data Analysis and Classification*, 1(3):255–291, 2007.
- [43] I. H. Witten and E. Frank. *Data Mining: Practical Machine Learning Tools and Techniques*. Elsevier, San Francisco, 2005.

MUSIC PATTERN DISCOVERY WITH VARIABLE MARKOV ORACLE: A UNIFIED APPROACH TO SYMBOLIC AND AUDIO REPRESENTATIONS

Cheng-i Wang, Jennifer Hsu and Shlomo Dubnov

Music Department

University of California, San Diego

{chw160, jsh008, sdubnov}@ucsd.edu

ABSTRACT

This paper presents a framework for automatically discovering patterns in a polyphonic music piece. The proposed framework is capable of handling both symbolic and audio representations. Chroma features are post-processed with heuristics stemming from musical knowledge and fed into the pattern discovery framework. The pattern-finding algorithm is based on *Variable Markov Oracle*. The *Variable Markov Oracle* data structure is capable of locating repeated suffixes within a time series, thus making it an appropriate tool for the pattern discovery task. Evaluation of the proposed framework is performed on the JKU Patterns Development Dataset with state of the art performance.

1. INTRODUCTION

Automatic discovery of musical patterns (motifs, themes, sections, etc.) is a task defined as identifying salient musical ideas that repeat at least once within a piece [3, 11] with computational algorithms. In contrast to “segments” found in the music segmentation task [14], the patterns found here may overlap with each other and may not cover the entire piece. In addition, the occurrences of these patterns could be inexact in terms of harmonization, rhythmic pattern, melodic contours, etc. Lastly, hierarchical relations between motifs, themes and sections are also desired outputs of the pattern discovery task.

Two major approaches for symbolic representations are the string-based and the geometric methods. A string-based method treats a symbolic music sequence as a string of tokens and applies string pattern discovery algorithms on the sequence [2, 18]. A geometric method views musical patterns as shapes appearing on a score and enables inexact pattern matching as similar shapes imply different occurrences of one pattern [4, 16]. For a comprehensive review of pattern discovery with symbolic representations, readers are directed to [11]. For audio representations, geometric

methods for symbolic representations have been extended to handle audio signals by multi F_0 -estimation with beat tracking techniques [5]. Approaches adopted from music segmentation tasks using self-similarity matrices and greedy search algorithms are proposed in [19, 20]. Most of the research involving audio representations has been focused on “deadpan audio” rendered from MIDI. In [5], the pattern discovery task is extended to live performance audio recordings with a single recording for each music piece. In the current study, instead of directly applying the proposed framework on performance recordings, multiple recordings are gathered for each musical piece to aid the pattern discovery on deadpan audio.

In this paper, the work presented in [25] focusing on pattern discovery on deadpan audio is extended to handle symbolic representations. The framework proposed in this paper can be seen as a string-based method in which input features are symbolized. The framework consists of two blocks: 1) feature extraction with post-processing routines and 2) the pattern finding algorithm. For both symbolic and audio representations, chroma features are extracted and post-processed based on musical heuristics, such as modulation, beat-aggregation, etc. The core of the pattern finding algorithm is a *Variable Markov Oracle (VMO)*. A *VMO* is a data structure capable of symbolizing a signal by clustering the observations in a signal, and is derived from the *Factor Oracle (FO)* [13] and *Audio Oracle (AO)* [9] structures. The *FO* structure is a variant of a suffix tree data structure and is devised for retrieving patterns from a symbolic sequence [13]. An *AO* is the signal extension of a *FO*, and is capable of indexing repeated sub-clips of a signal sampled at discrete times. *AOs* have been applied to audio query [6] and audio structure discovery [8]. The *VMO* data structure was first proposed in [24] as an efficient audio query-matching algorithm. This paper shows the capability of using a *VMO* to find repeated sub-clips in a signal in an unsupervised manner.

This paper is structured as follows: section 2 introduces the *VMO* data structure and the accompanying pattern finding algorithm. Section 3 documents the experiments on symbolic and audio representations as well as the dataset, feature extraction, and task setup. Section 4 provides an evaluation of the experiment. Last, future work, observations and insights are discussed in section 5.



© Cheng-i Wang, Jennifer Hsu and Shlomo Dubnov.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Cheng-i Wang, Jennifer Hsu and Shlomo Dubnov. “Music Pattern Discovery with Variable Markov Oracle: A Unified Approach to Symbolic and Audio Representations”, 16th International Society for Music Information Retrieval Conference, 2015.

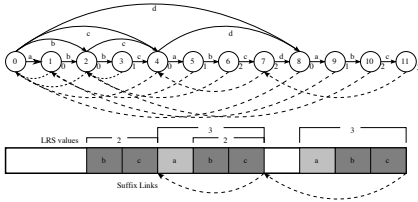


Figure 1. (Top) A *VMO* structure with symbolized signal $\{a, b, c, a, b, c, d, a, b, c\}$, upper (solid) arrows represent forward links with symbols for each frame and lower (dashed) are suffix links. Values outside of each circle are the lrs value for each state. (Bottom) A visualization of how patterns $\{a, b, c\}$ and $\{b, c\}$ are related to lrs and sfx .

2. VARIABLE MARKOV ORACLE

A *VMO* symbolizes a time series O , sampled at time t , into a symbolic sequence $Q = q_1, q_2, \dots, q_t, \dots, q_T$, with T states and with frame $O[t]$ labeled by a symbol q_t . The symbols are formed by tracking suffix links along the states in an oracle structure. An oracle structure (either *FO*, *AO* or *VMO*) carries three kinds of links: forward link, suffix link and reverse suffix link. A suffix link is a backward pointer that links state t to k with $t > k$, without a label, and is denoted by $sfx[t] = k$.

$$sfx[t] = k \iff \text{the longest repeated suffix of } \{q_1, q_2, \dots, q_t\} \text{ is recognized in } k.$$

Suffix links are used to find repeated suffixes in Q . In order to track the longest repeated suffix at each time index t , the length of the longest repeated suffix at each state t (denoted as $lrs[t]$) is computed by the algorithm described in [13]. A reverse suffix link, $rsfx[k] = t$, is the suffix link in the reverse direction. sfx , lrs and $rsfx$ allow for the proposed pattern discovery algorithm described in section 2.2.

Forward links are links with labels and are used to retrieve any of the factors from Q . Since forward links are not used in the proposed algorithm, readers are referred to [13] for details.

The last piece for the construction of a *VMO* is a threshold value, θ . θ is used to determine if the incoming $O[t]$ is similar to one of the frames following the suffix link beginning at $t - 1$. Two frames, $O[i]$ and $O[j]$, are assigned the same symbol if $|O[i] - O[j]| \leq \theta$. In extreme cases, a *VMO* may assign different symbols to every frame in O (θ excessively low), or a *VMO* may assign the same symbol to every frame in O (θ excessively high). In these two cases, the *VMO* structure is incapable of capturing any patterns (repeated suffixes) in the signal. The optimal θ can be found by calculating the *Information Rate (IR)*, a music information dynamics measure, and this process is described in section 2.1. An example of an oracle structure with extreme θ values is shown in Fig. 2.

The on-line construction algorithms of *VMO* are intro-

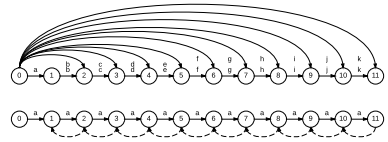


Figure 2. Two oracle structures with extreme values of θ . The characters near each forward link represent the assigned labels. (Top) The oracle structure with $\theta = 0$ or extremely low θ value. (Bottom) The oracle structure with a very high θ value. In both cases the oracles are not able to capture any structure in the time series.

duced in [24] and not repeated here. Fig. 1 shows an example of a constructed *VMO* and how lrs and sfx are related to pattern discovery. The symbols formed by gathering states connected by suffix links share the following properties: 1) the pairwise distance between states connected by suffix links is less than θ , 2) the symbolized signal formed by the oracle can be interpreted as a sample from a variable-order Markov model because the states connected by suffix links share common suffixes with variable length, 3) each state is labeled by a single symbol because each state has a single suffix link, 4) the alphabet size of the assigned symbols is unknown before the construction and is determined by θ .

2.1 Model Selection via Information Rate

The same input signal may be associated with multiple *VMOs* with different suffix structures and different symbolized sequences if different θ values are used to construct the *VMOs*. To select the one symbolized sequence with the most informative patterns, *IR* is used as the criterion in model selection between different structures generated by different θ values. *IR* is an information theoretic measure capable of measuring the information content of a time series [7] in terms of the predictability of its source process on the present observation given past ones. In the context of pattern discovery with a *VMO*, a *VMO* with higher *IR* value captures more of the repeating sub-clips (ex. patterns, motives, themes, gestures, etc) than the ones with lower *IR* values.

The *VMO* structure uses the same approach as the *AO* structure [8] to calculate *IR*. Let $x_1^N = \{x_1, x_2, \dots, x_N\}$ denote time series x with N observations, $H(x)$ the entropy of x , the definition of *IR* is

$$IR(x_1^{n-1}, x_n) = H(x_n) - H(x_n | x_1^{n-1}). \quad (1)$$

IR is the mutual information between the present and past observations and is maximized when there is a balance between variations and repetitions in the symbolized signal. The value of *IR* can be approximated by replacing the entropy terms in (1) with complexity measures associated with a compression algorithm. These complexity measures

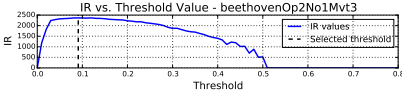


Figure 3. IR values are shown on the vertical axis while θ are on the horizontal axis. The solid blue curve shows the relationship between IR and θ , and the dashed black line indicates the chosen θ by locating the maximum IR value. Empirically, IR curves exhibit quasi-concave function shapes, thus a global maximum can be located.

Algorithm 1 Pattern Discovery using VMO

Require: VMO , V , of length T and minimum pattern length L .
Ensure: sfx , $rsfx$, $lrs \in V$

- 1: Initialize $Pptr$ and $PptrLen$ as empty lists.
- 2: Initialize $prevSfx = -1$, $K = 0$
- 3: **for** $i = T : L$ **do**
- 4: $pPtrFound = False$
- 5: **if** $i - lrs[i] + 1 > sfx[i] \wedge sfx[i] \neq 0 \wedge lrs[i] \geq L$ **then**
- 6: **if** $\exists k \in \{1, \dots, K\}$, $sfx[i] \in Pptr[k]$ **then**
- 7: Append i to $Pptr[k]$
- 8: $PptrLen[k] \leftarrow \min(lrs[i], PptrLen[k])$
- 9: $pPtrFound = True$
- 10: **end if**
- 11: **if** $prevSfx - sfx[i] \neq 1 \wedge pPtrFound == False$ **then**
- 12: Append $\{sfx[i], i, rsfx[i]\}$ to $Pptr$
- 13: Append $\min\{lrs\{\{sfx[i], i, rsfx[i]\}\}$ to $PptrLen$
- 14: $K \leftarrow K + 1$
- 15: **end if**
- 16: $prevSfx \leftarrow sfx[i]$
- 17: **else**
- 18: $prevSfx \leftarrow -1$
- 19: **end if**
- 20: **end for**
- 21: **return** $Pptr, PptrLen, K$

are the number of bits used to compress x_n independently and compress x_n using the past observations x_{n-1}^n . The formulation of combining the lossless compression algorithm, *Compr* [12], with *AO* and *IR* is provided in [8]. A visualization of the sum of IR values versus different θ s on one of the music pieces tested in this paper is depicted in Fig. 3.

2.2 Pattern Discovery

Algorithm 1 shows the *VMO*-based algorithm for the automatic pattern discovery task. The idea behind Algorithm 1 is to track patterns by following sfx and lrs . sfx provides the locations of patterns, and lrs indicates the length of these patterns. In line 5 of Algorithm 1, checks are made so that redundant patterns are avoided, and the lengths of patterns are larger than a user-defined minimum L . From line 6 to 10, the algorithm recognizes occurrences of established patterns, and from line 11 to 15 it detects new patterns and stores them into $Pptr$ and $PptrLen$.

Algorithm 1 returns $Pptr, PptrLen$ and K . $Pptr$ is a list of lists with each $Pptr[k], k \in \{1, 2, \dots, K\}$, a list containing the ending indices of different occurrences of the k th pattern found. K is the total number of patterns found. $PptrLen$ has K values representing the length of the k th pattern in $Pptr$.

3. EXPERIMENTS

The dataset chosen for the music pattern discovery is the JKU Pattern Development Dataset (JKU-PDD) [3]. This dataset consists of five polyphonic classical music pieces or movements in both symbolic and audio representations. The ground truth of repeated patterns (motifs, themes, sections) for each piece is annotated by musicologists. The details of the experimental setup are provided in the following sections.

3.1 Feature Extraction

For the automatic musical pattern discovery task, the chromagram is the input feature to Algorithm 1 for both the symbolic and audio representations. The chromagram is a feature that characterizes harmonic content and is a commonly used in musical structure discovery [1].

3.1.1 Symbolic Representation

For the experiments described in this paper, the symbolic representation chosen is MIDI, but other symbolic representations may be used instead. The chromagram derived from the symbolic representation is referred to as the “MIDI chromagram”.

The MIDI chromagram is similar to the MIDI histogram described in [23] and represents the presence of pitch classes during each time frame. To create a MIDI chromagram with quantization b in terms of MIDI whole note beats, frame size M , and hop size h , the MIDI file is first parsed into a matrix where each column is a MIDI beat quantized by b and each row is a MIDI note number ($0 - 127$). For each analysis frame, the velocities are summed over M MIDI beats, and then folded and summed along the MIDI notes to create a single octave of velocities. In other words, all velocities that correspond to MIDI notes that share the same modulo 12 are summed. The analysis frame then hops h MIDI beats forward in time, repeats the folding and summing, and continues on until the end of the MIDI matrix is reached. The bottom plot in Fig. 4 is an example of the MIDI chromagram extracted from the Beethoven minuet in the JKU-PDD.

3.1.2 Audio Recording

The routines for extracting the chromagram from an audio recording used in this paper is as follows. For a mono audio recording sampled at 44.1 kHz, the recording is first downsampled to 11025 Hz. Next, a spectrogram is calculated using a Hann window of length 8192 with 128 samples overlap. Then the constant-Q transform of the spectrogram is calculated with frequency analysis ranging between $f_{min} = 27.5$ Hz to $f_{max} = 5512.5$ Hz and 12 bins per octave. Finally, the chromagram is obtained by folding the constant-Q transformed spectrogram into a single octave to represent how energy is distributed among the 12 pitch classes.

To achieve the pattern discovery on a music metrical level, the chroma frames are aggregated with a median filter according to the beat locations found by a beat tracker

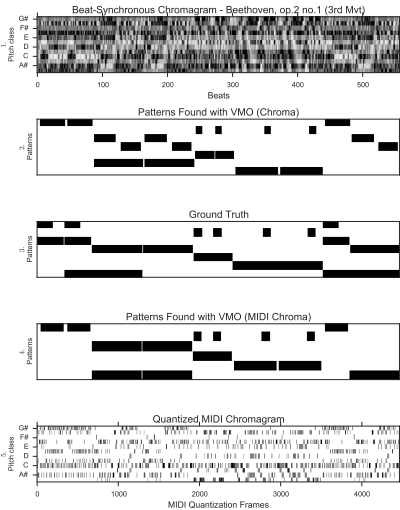


Figure 4. Features, found patterns, and ground truth for the Beethoven minuet in the JKU-PDD. 1. Beat-synchronous chromagram from the deadpan audio recording. 2. Patterns found by Algorithm 1 using the chromagram shown above. 3. Ground truth from JKU-PDD. 4. Patterns found by Algorithm 1 using the MIDI chromagram. 5. Quantized MIDI chromagram. For 2., 3. and 4., each row is a pattern place holder with dark regions representing the occurrences on the timeline. The order of found patterns is manually sorted to best align with the ground truth for visualization purpose. Notice the hierarchical relations of patterns embedded in the ground truth and found from the algorithms.

[10] conforming to the music metrical grid. For finer rhythmic resolution, each beat identified is spliced into two sub-beats before chroma frame aggregation. Last, the sub-beat-synchronous chromagram is whitened with a \log function. Whitening boosts the harmonic tones implied by the motifs so that the difference between the same motif with and without harmonization is reduced. See the top plot in Fig. 4 for an example of the the beat-synchronous chromagram extracted from the Beethoven minuet in the JKU-PDD.

3.2 Repeated Themes Discovery

For both symbolic and audio representations, after the chroma feature sequence O is extracted from the music piece as described in section 3.1.1 and 3.1.2, $\theta \in (0.0, 2.0]$ is used to construct multiple VMO s with O . The L_2 -norm is used to calculate the distance between incoming observations and the ones stored in a VMO . The single VMO with the highest IR is fed into Algorithm 1 with L to find patterns and their occurrences. Instead of setting $L = 5$

for all pieces as in [25], L is set according to lrs as $L = \frac{\gamma}{T} \sum_{t=1}^T \text{lrs}[t]$, where L is adaptive to the average length of repeated suffixes found in the piece. γ is a scaling parameter which is set to 0.5 empirically.

To consider transposition (moving patterns up or down by a constant pitch interval), the distance function used for VMO structures is a cost function with transposition invariance. For a transposition invariant cost function, a cyclic permutation with offset k on an n -dimensional vector $\mathbf{x} = (x_0, x_1, \dots, x_{n-1})$ is defined as

$$cp_k(\mathbf{x}) := \{x_i \rightarrow x_{(i+k \bmod n)}, \forall i \in (0, 1, \dots, n-1)\},$$

and the transposition invariant dissimilarity d between 2 vectors x and y is defined as, $d = \min_k \{\|x - cp_k(y)\|_2\}$. $n = 12$ for the chroma vector, and the cost function is used during the VMO construction.

In addition to the basic chromagram, a stacked chromagram using time-delay embedding with M steps of history as in [22] is also used. Experiments reveal that choices for b , M , and h for both the MIDI chromagram and the stacked MIDI chromagram can greatly alter the accuracy of patterns discovered. The values used in the experiments were quantization sizes $b = [\frac{1}{8}, \frac{1}{16}, \frac{1}{32}]$, frame size $M = [1, 8, 16, 32]$, and hop lengths $h = [1, 2, 4]$ where M and h are described in terms of MIDI beats of size b . It was found that the stacked MIDI chromagram with $b = \frac{1}{32}$, $M = 16$, and $h = 2$ resulted in the best pattern discovery. For the audio representation, there is no significant difference in terms of the patterns found or the evaluation metrics between regular and stacked chromagrams.

Fig. 4 shows the chromagram found from audio and MIDI for the Beethoven minuet in the JKU-PDD along with the patterns found by the VMO structure and the ground truth patterns. The patterns found by the audio and symbolic representations share similarities and visually resemble the ground truth patterns. In section 4, quantitative measures for evaluating the patterns found by the VMO are explained and reported.

3.3 Performance Recordings to Aid Pattern Discovery

Five performance recordings for each of the pieces included in the JKU-PDD are collected in order to further explore the discovery of repeated themes. The motivation behind this experiment is to explore the notion that music performances contain information about how performers interpret the musical structure embedded in the score [21] and to examine whether or not the patterns found on deadpan audio could be improved with the addition of such information.

For each of the performance recordings, the chromagram is extracted and aggregated along the beats as described in section 3.1.2. Dynamic Time Warping [17] is used to align the beat-synchronous chromagram from the performance audio with the beat-synchronous chromagram of the deadpan audio. Since motif annotations on these performance recordings do not exist yet, the alignment between the deadpan audio and performance recordings are necessary so that the patterns found from the performance

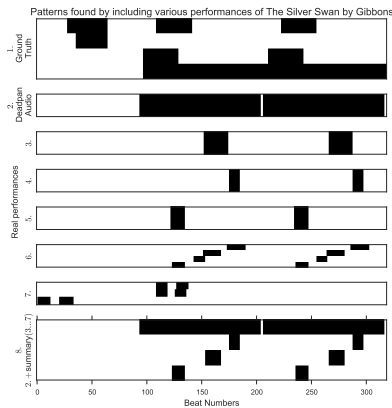


Figure 5. 1. Ground truth from JKU-PDD. 2. Patterns found from deadpan audio with the *VMO*. 3 – 7. Patterns found from the five performances. 8. Patterns from deadpan and performance audio.

recordings can be compared to the ground truth or added to the found patterns from the deadpan audio. The drawback of the alignment is that timing variations containing the performer’s structural interpretation are lost. Although timing variations are lost in this experiment, velocity variations applied across time and different voices are retained. The aligned performance audio chromagram is then whitened, normalized and fed into the *VMO* pattern finding algorithm. For patterns found across multiple performances of one piece, the intersection of patterns for any two performances of one piece that are longer than L are kept and added to the found patterns from the deadpan audio. Fig. 5 is an example of how incorporating performance recordings can change the discovered patterns from deadpan audio.

4. EVALUATION

The evaluation follows the metrics proposed in the Music Information Retrieval Evaluation eXchange (MIREX) [3]. Three metrics are considered for inexact pattern discovery. For each metric, standard F_1 score, defined as $F_1 = \frac{2PR}{P+R}$, precision P and recall R are calculated. The first metric is the establishment score (*est*) which measures how each ground truth pattern is identified and covered by the algorithm. The establishment score takes inexactness into account and does not consider occurrences. The second metric is the occurrence score (*oc*) with a threshold c . The occurrence score measures how well the algorithm performs in finding occurrences of each pattern. The threshold c determines whether or not an occurrence should be counted. The higher the value for c , the lower the tolerance. $c = \{0.5, 0.75\}$ are used in standard MIREX

evaluation. The last metric is the three-layer score that considers both the establishment and occurrence score. The results of the proposed framework are listed in Table 1 along with a comparison to previous work.

From the evaluations for both symbolic and audio representations, the establishment scores are generally lower than the occurrence scores, meaning that the proposed algorithm is better at finding occurrences of established patterns than finding all possible patterns. With the symbolic representation, the standard F_{est} , $F_{oc(0.75)}$, and F_3 scores are better than previously published results. The establishment, occurrence, and three-layer precision scores are also as good as or better than previous algorithms [5, 15]. The recall scores reveal that this is a part of the algorithm that could be improved as previous algorithms all scored higher on recall than the proposed algorithm. Similar to the symbolic results, the proposed audio algorithm achieves high F_1 and precision scores for the establishment, occurrence, and three-layer scores. The recall of the audio algorithm is higher than previously reported results [5, 19, 20]. The recall rates of the proposed framework are inferior when compared to the precision scores and previous work in symbolic representation. This may occur because chroma features were used and the folding of the constant-Q spectrogram discards information contained in different voices.

The inclusion of performance recordings is the effort made in this work to improve both the coverage and accuracy of the pattern discovery framework for audio representations. Due to space limitations, the detailed metrics for each piece in the JKU-PDD is not shown here. The effects of including performance recordings are described here. The establishment recall rate and occurrence precision rate with threshold 0.5 are improved when performance recordings are included, but in general the pattern discovery task is not improved because the decrease in establishment precision rate is larger than the improvement on recall rates. This result indicates that more patterns and their occurrences could be discovered if different versions of the same piece are used in the pattern discovery task, but more false positive patterns will be found.

The proposed pattern finding algorithm completed in less time than previously reported algorithms on both symbolic and audio representations. Although the *VMO* data structure is used for both the proposed symbolic and audio algorithms, there is a discrepancy in the time that it takes to find the patterns for all five songs. The audio algorithm takes much less time because the analysis frames are larger than the frames used in the symbolic representation (32th note versus 8th note relatively). Thus, there are less frames to analyze with the audio representation and building a *VMO* takes less time.

Fig. 6 is a summary of the three-layer F_1 scores for each of the 5 pieces in the JKU-PDD for the proposed audio and symbolic frameworks along with the current state of the art results. The small quantization value for the MIDI representation leads to a higher score in the case of the Beethoven and Chopin pieces. The proposed audio

Algorithm	P_{est}	P_{est}	R_{est}	$F_{ol(.5)}$	$P_{ol(.5)}$	$R_{ol(.5)}$	$F_{ol(.75)}$	$P_{ol(.75)}$	$R_{ol(.75)}$	F_3	P_3	R_3	Time (s)
VMO symbolic	60.79	74.57	56.94	71.92	79.54	68.78	75.98	75.98	75.99	56.68	68.98	53.56	4333
[5]	33.7	21.5	78.0	76.5	78.3	74.7	—	—	—	—	—	—	—
[15]	50.20	43.60	63.80	63.20	57.00	71.60	68.40	65.40	76.40	44.20	40.40	54.40	7297
VMO deadpan	56.15	66.8	57.83	67.78	72.93	64.3	70.58	72.81	68.66	50.6	61.36	52.25	96
deadpan + real	52.76	53.2	58.25	67.35	74.42	63.31	70.51	72.73	68.58	48.25	50.2	52.84	—
[20]	49.8	54.96	51.73	38.73	34.98	45.17	31.79	37.58	27.61	32.01	35.12	35.28	454
[5]	23.94	14.9	60.9	56.87	62.9	51.9	—	—	—	—	—	—	—
[19]	41.43	40.83	46.43	23.18	26.6	20.94	24.87	32.08	21.24	28.23	30.43	31.92	196

Table 1. Results from various algorithms on the JKU-PDD for both symbolic (upper three) and audio (bottom four) representations. Scores are averaged across pieces. Missing values were not reported in their original publications.

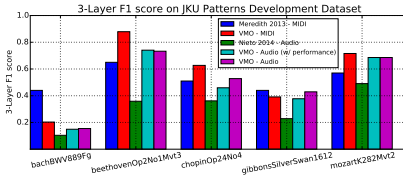


Figure 6. Three-layer F_1 score (F_3 in Table 1) for the proposed audio and symbolic method on the 5 pieces in the JKU-PDD plotted along with state of the art results.

and symbolic framework have the highest F_1 value on the Beethoven minuet and the lowest F_1 value with the Bach Fugue. When looking at the proposed method along with current the state of the art results, it is evident that the Bach Fugue and the Gibbons piece are songs where patterns are embedded in different voices, and that the Beethoven piece has more consistent repeated phrases. The algorithm for symbolic data described in [15] performs better with Bach and Gibbons in comparison to VMO and [20], most likely because of its capability to discover patterns embedded in different yet simultaneous voices.

In summary, our method has improved upon the F_1 and P scores as well as time to find patterns. The patterns found using audio and symbolic representations are similar and the evaluation scores reflect this similarity. Improving recall and allowing for inexact occurrences should be a focus for future studies. Source codes and details about the experiments are accessible via Github¹.

5. DISCUSSION

In this work, a framework for automatic pattern discovery from a polyphonic music piece based on a VMO is proposed and shown to achieve state of the art performance on the JKU-PDD dataset. With both the regular and stacked MIDI chromagram, a smaller quantization value b results in better pattern discovery because finer details are captured with smaller quantization. From the results, it seems that a larger frame size M for smaller quantization b resulted in better pattern finding. For hop size h , it is observed that $h = 2$ results in a hop of a 16th note which

is the shortest note in the JKU-PDD ground truth annotations. Results from both the audio and MIDI representations show that the recall of discovered themes could be improved. Although it is possible for a VMO to identify inexact patterns from the input feature sequence with symbolization from θ , different occurrences of the same pattern are sometimes not recognized because chroma features discard information from various voices in the music piece. Our framework could be improved if the feature used allows for separation of voices from polyphonic MIDI and audio. Incorporating techniques for identifying multiple voices in polyphonic music would improve the proposed framework.

In addition to the proposed framework for both symbolic and audio representations, using multiple performance recordings in the repeated themes discovery task for deadpan audio is another novelty presented in this paper. The work done in this paper differs from [5] in that the performance audio recordings are used as supplements to deadpan audio and not analyzed as separate musical entities. The original intention behind using deadpan audio for repeated themes discovery is to allow for the use of audio signal processing techniques, but deadpan audio contains the same amount of information as its symbolic counterpart with less accessibility because of its representation. This is evident by the similarity between the MIREX metrics for the MIDI and deadpan audio since similar techniques are applied. Performance recordings, on the other hand, contain expressive performance variations on phrasing and segmentation. In this paper, it is shown that adding performance recordings to the proposed framework achieved improvements on some of the standard metrics. The next step for advancing the repeated themes discovery task is to annotate the performance recordings so that these recordings can be used as a dataset directly without referencing back to the deadpan audio version. By observing the results from the pattern finding with performance recordings, the patterns found for each performance show informative cues as to how each rendition of the same piece differs from the others visually (Fig. 5). These visualizations are interesting discoveries on their own, even without a comparison to ground truth annotations, and could be further investigated for use in expressive performance analysis and music structural segmentation.

¹ https://github.com/wangsix/VMO_repeated_themes_discovery

6. REFERENCES

- [1] Juan Pablo Bello. Measuring structural similarity in music. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(7):2013–2025, 2011.
- [2] Emiliós Cambouropoulos, Maxime Crochemore, Costas S Iliopoulos, Manal Mohamed, and Marie-France Sagot. All maximal-pairs in step-leap representation of melodic sequence. *Information Sciences*, 177(9):1954–1962, 2007.
- [3] Tom Collins. Discovery of repeated themes and sections. Retrieved 4th May, http://www.music-ir.org/mirex/wiki/2013:Discovery_of_Repeated_Themes_&_Sections, 2013.
- [4] Tom Collins, Andreas Arzt, Sebastian Flossmann, and Gerhard Widmer. SIARCT-CFP: Improving precision and the discovery of inexact musical patterns in point-set representations. In *ISMIR*, pages 549–554, 2013.
- [5] Tom Collins, Sebastian Böck, Florian Krebs, and Gerhard Widmer. Bridging the audio-symbolic gap: The discovery of repeated note content directly from polyphonic music audio. In *Audio Engineering Society Conference: 53rd International Conference: Semantic Audio*. Audio Engineering Society, 2014.
- [6] Arshia Cont, Shlomo Dubnov, Gérard Assayag, et al. Guidage: A fast audio query guided assemblage. In *International Computer Music Conference*, 2007.
- [7] Shlomo Dubnov. Spectral anticipations. *Computer Music Journal*, 30(2):63–83, 2006.
- [8] Shlomo Dubnov, Gérard Assayag, and Arshia Cont. Audio oracle analysis of musical information rate. In *Semantic Computing (ICSC), 2011 Fifth IEEE International Conference on*, pages 567–571. IEEE, 2011.
- [9] Shlomo Dubnov, Gerard Assayag, Arshia Cont, et al. Audio oracle: A new algorithm for fast learning of audio structures. In *International Computer Music Conference*, 2007.
- [10] Daniel P. W. Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, 2007.
- [11] Berit Janssen, W. Bas de Haas, Anja Volk, and Peter Kraneburg. Discovering repeated patterns in music: potentials, challenges, open questions. In *10th International Symposium on Computer Music Multidisciplinary Research*. Laboratoire de Mécanique et d’Acoustique, 2013.
- [12] Arnaud Lefebvre and Thierry Lecroq. Compror: online lossless data compression with a factor oracle. *Information Processing Letters*, 83(1):1–6, 2002.
- [13] Arnaud Lefebvre, Thierry Lecroq, and Joël Alexandre. An improved algorithm for finding longest repeats with a modified factor oracle. *Journal of Automata, Languages and Combinatorics*, 8(4):647–657, 2003.
- [14] Brian McFee and Daniel P. W. Ellis. Analyzing song structure with spectral clustering. In *The 15th International Society for Music Information Retrieval Conference*, pages 405–410, 2014.
- [15] David Meredith. COSIATEC and SIATECCompress: Pattern discovery by geometric compression. In *International Society for Music Information Retrieval Conference*, 2013.
- [16] David Meredith, Kjell Lemström, and Geraint A. Wiggins. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, 31(4):321–345, 2002.
- [17] Meinard Müller. Dynamic time warping. *Information retrieval for music and motion*, pages 69–84, 2007.
- [18] Oriol Nieto and Morwaread Farbood. Perceptual evaluation of automatically extracted musical motives. In *Proceedings of the 12th International Conference on Music Perception and Cognition*, pages 723–727, 2012.
- [19] Oriol Nieto and Morwaread Farbood. MIREX 2013: Discovering musical patterns using audio structural segmentation techniques. *Music Information Retrieval Evaluation eXchange, Curitiba, Brazil*, 2013.
- [20] Oriol Nieto and Morwaread Farbood. Identifying polyphonic patterns from audio recordings using music segmentation techniques. In *The 15th International Society for Music Information Retrieval Conference*, 2014.
- [21] John Rink, Neta Spiro, and Nicolas Gold. Motive, gesture, and the analysis of performance. *New Perspectives on Music and Gesture*, pages 267–292, 2011.
- [22] Joan Serrà, Meinard Mueller, Peter Grosche, and Josep Ll Arcos. Unsupervised music structure annotation by time series structure features and segment similarity. *IEEE Transactions on Multimedia*, 16(5):1229–1240, 2014.
- [23] George Tzanetakis, Andrey Ermolinskyi, and Perry Cook. Pitch histograms in audio and symbolic music information retrieval. *Journal of New Music Research*, 32(2):143–152, 2003.
- [24] Cheng-i Wang and Shlomo Dubnov. Guided music synthesis with variable markov oracle. In *The 3rd International Workshop on Musical Metacreation, 10th Artificial Intelligence and Interactive Digital Entertainment Conference*, 2014.
- [25] Cheng-i Wang and Shlomo Dubnov. Pattern discovery from audio recordings by variable markov oracle: A music information dynamics approach. In *Acoustics, Speech, and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015.

AUTOMATIC SOLFÈGE ASSESSMENT

Rodrigo Schramm¹ Helena de Souza Nunes² Cláudio Rosito Jung¹

¹ Institute of Informatics, Federal University of Rio Grande do Sul, Brazil

² Department of Music, Federal University of Rio Grande do Sul of Music, Brazil

rodrigoss@caef.ufrgs.br, helena@caef.ufrgs.br, crjung@inf.ufrgs.br

ABSTRACT

This paper presents a note-by-note approach for automatic solfège assessment. The proposed system uses melodic transcription techniques to extract the sung notes from the audio signal, and the sequence of melodic segments is subsequently processed by a two stage algorithm. On the first stage, an aggregation process is introduced to perform the temporal alignment between the transcribed melody and the music score (ground truth). This stage implicitly aggregates and links the best combination of the extracted melodic segments with the expected note in the ground truth. On the second stage, a statistical method is used to evaluate the accuracy of each detected sung note. The technique is implemented using a Bayesian classifier, which is trained using an audio dataset containing individual scores provided by a committee of expert listeners. These individual scores were measured at each musical note, regarding the pitch, onset, and offset accuracy. Experimental results indicate that the classification scheme is suitable to be used as an assessment tool, providing useful feedback to the student.

1. INTRODUCTION

The practice of solfège is used by beginner musicians to learn and improve the ability of the musical reading through the repeated singing of musical notes from a music score. In fact, this kind of exercise is a fundamental part of the music learning process. It guides the student to build its own musical perceptions by creating an internal image of the sound along the vocal emission of a note (or sequences of notes as intervals, scales and melodies). The ability to read the notes on a music score and at the same time to hear internally and to sing them *a prima vista* is here generically called solfège, and it is considered a prerequisite for performance and effective musical knowledge [13]. During the solfège, is crucial to have a constant feedback by an external expert, who should be responsible for detecting eventual mistakes and pinpoint the best way to fix them. Traditionally, the evaluation process of the solfège is conducted by a music teacher, inside of a classroom. Nowadays, with the spread

of the internet, new educational methods bring up new possibilities to music education using the e-learning paradigm. In the case of large number of evaluations, which is a typical situation in distance learning courses, the labor of the teacher becomes exhaustive and tedious. Even in cases of traditional and presential music lessons, the judgment by an expert musician is not a trivial task, specially because the human discernment may be affected by subjective factors and fatigue [14,20]. Thus, an automatic solfège assessment tool can be very helpful in this context.

Usually, solfège is evaluated by comparing the singing performance with the target music score (ground truth). In this case, the first part of this task have some similarities with automatic melodic transcription algorithms [14]. However, a set of similarity measures to correlate the user performance with the expert's (human) judgment is still needed. Although there are some papers that provide an overall score for a given solfège [10], it is not to our knowledge the existence of systems that perform a note-by-note analysis, which is very important in music teaching. In this paper we introduce a new note-by-note evaluation method based on the individual scores provided by human evaluators. More precisely, we introduce a Bayesian classifier which is applied to each musical note detection, working as an alternative to the correlation method based on the global judgment score used in [10]. The main difference here is the fact that the performance can be evaluated with a small granularity, at each musical note, but keeping the assessment correlated with the human judgment. Additionally, the Bayesian approach allows the mapping of the performance errors into a confidence measure. We also introduced a new temporal alignment method between the transcribed melody and the music score (ground truth) by using a clustering process. The grouping process was chosen in place of dynamic time warping (DTW) [12] approach because it is less sensible to error propagation and it does not have any monotonicity condition.

This paper is organized as follows: Section 2 presents an overview about the related techniques. Section 3 shows a detailed description of the audio database generation and the corresponding annotation process by the musicians experts. Section 4 describes the proposed method to automatic solfège assessment. Section 5 presents the results of our experiments and Section 6 draws the conclusion of this work.



© Rodrigo Schramm, Helena de Souza Nunes, Cláudio Rosito Jung. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Rodrigo Schramm, Helena de Souza Nunes, Cláudio Rosito Jung. "Automatic Solfège Assessment", 16th International Society for Music Information Retrieval Conference, 2015.

2. RELATED WORKS

As far as we know, there is no method for solfège evaluation in a note-by-note scale. Therefore, this section will revise some papers that tackle related problems. For example, Jha and Rao [4] focused on the vowel quality of the singing voice. The authors use low-level features, including the spectrum envelope and pitch contour for singing evaluation. Their algorithm detects the onset of each vowel by searching for rapid changes in specific frequency bands that characterize the vowel formants, and then correlates each vowel with an articulatory space by a linear regression scheme. Miryala et al. [8] do not perform assessment directly, but their approach automatically identifies vocal expressions as voice glides and vibratos, which could be also used as a kind of singing evaluation.

The related problem of melodic transcription has been studied by several researchers. The common pipeline on the melody transcription techniques splits the process in low-level feature estimation, note segmentation and labeling, and post processing [3, 11]. For example, [19] implemented a melodic transcription algorithm by detecting a sequence of fundamental frequencies in a frame-wise fashion, which are subsequently converted into observation probabilities and used in a Hidden Markov Model (HMM). Ryyanen and Klapuri [17] implemented a similar approach, but extending the number of low-level features. Thus, besides the fundamental frequency estimates, they also mapped into probabilities distributions the features regarding voice/unvoice, accent, and meter estimation. Frequently, a musicological model is also included in music transcription algorithms to improve the system accuracy, acting as a prior probability. The authors of [19] also incorporate a duration model, which maps probability density functions with the subdivisions and multiples unities of the beat time. Musicological models might be used to detect the tonality and the rhythmic structure of the musical performance, constraining the output options and consequently improving the accuracy [5]. Unfortunately, the musicological model cannot be directly used as *a priori* information on assessment tools since it is not possible to have any expectation about the student singing performance.

The work by Molina et al. [10], which explores the singing assessment regarding note-based melodic similarities, as well as the temporal alignment between the student performance and the target melody, has similar goals to ours. Despite the use of note-level similarity measures, the final evaluation is built using the global assessment scores from the human experts, who had placed a global score (between 1 and 10) for each singing performance on a previous training stage. The estimated correlations in [10] seem to be advantageous to extract a measure of quality in a global context. However, as a drawback, that approach discard local (note level) information from the experts' evaluation. In other words, it is not possible to precisely locate and quantify the note(s) responsible for a bad or good score from the singing performance. A small extension of this approach was presented in [6], including new audio spectral features. A recent work [9] shows a taxonomy of evaluation

measures used in several automatic singing transcriptions algorithms. Most of the tabulated approaches have used evaluation measures for singing transcription algorithms based on note/frame-level error. There are also some strategies that use time warping alignment information between the ground truth and the transcribed melody [10]. Despite the variety and effort to build robust and comprehensive evaluations measures, these previous ideas cannot be directly used in the context of solfège assessment. In fact, the used definition of correct pitch/ onset/offset in [9] applies ranges of tolerance with fixed values, that may be a reliable procedure to compare distinct algorithms of melodic transcription. However, it may not agree with the human judgment perception in a solfège assessment context. Some authors [6, 10] tried to solve this issue connecting the expert analysis with the evaluation measures, but the final human evaluation carries out only a global interpretation, lacking in details at individual sung notes. In the next sections we present our dataset and proposed model for solfège assessment. This model aims to evaluate individual sung notes, giving a note-based feedback that makes a meaningful link with the human judgment by musician experts.

3. PROPOSED DATASET AND ANNOTATIONS

The proposed dataset consists of sequences of musical intervals in the chromatic scale. The audio recordings were done using seven adults, including trained (three) and untrained (four) singers ranging from 17 to 61 years old. These melodic sequences were recorded during four months, in mono format with a sample rate of 44100Hz and 16 bits quantization.

It was decided to support the singing process by a reference piano audio track, since a part of the group of singers was unable to read music scores. In this reference audio track, the intervals were played in sequence, but with gaps between them. Each singer filled these gaps repeating the previous heard melodic interval at the next beat time, and all recording sessions were synchronized by a metronome.

The singers were asked to choose and, if possible, to diversify the used phonemes. They were also asked to sing freely, but respecting the pitch, attack and duration of the previously indicated sounds, aiming to capture real examples of spontaneous everyday singing. Intentionally aiming to capture a higher variability of natural situations, the recordings were conducted in two distinct environments: a part of the examples was recorded in a studio, where the resulting audio records are clean; another part of the audio records was done in informal conditions, presenting some background noise and reverberation.

A total of 21 sessions were recorded, containing (twelve ascending intervals and twelve descendants intervals of the chromatic scale). Each singer performed the melodic intervals in three distinct tempos: Adagio, 60 bpm; Andante Moderato, 90 bpm; and Allegro, 120 bpm. Along with the recordings, an annotation process was conducted by a committee of experts (five graduated musicians with more than ten years of experience in solfège assessment auditions) in order to label each sung note from the recorded dataset into

two possible categories (correct and incorrect) regarding the pitch, onset and offset accuracy. Before each annotation section, the committee was advised to hear some random samples from the dataset. This warmup procedure was important because it helped to create an agreement among the experts, who shared some important characteristics and aspects of the recorded melodic intervals. As the dataset was broken in parts, this process was repeated in several days, until the whole set of audio records had been evaluated (in fact, the whole process for building the annotated dataset took several months).

For each sung note in the audio dataset, all the five evaluators casted a vote (correct or incorrect) regarding each analyzed parameter (pitch, onset and offset). As it will be explained in the next section, disagreement among the evaluators were kept and used to model our probabilistic classifier. Also, each note is assigned to a single label (correct or incorrect) regarding to each parameter, based on the majority of votes cast by the experts (i.e., at least 3 votes for the same label). Hence, some labels can be considered more reliable than others, based on the number of votes. For example, regarding the pitch, 15.38% of the samples received 3 votes in agreement, which means an expressive degree of doubt among the experts. The same analysis was made for the onset and offset parameters, and the percentage of notes with 3 votes (doubt) was 10.71% and 12.09%, respectively. The final annotated dataset contains 3276 labeled samples.

4. OUR MODEL

The proposed computational model for automatic solfège evaluation is structured in two main stages. The first stage performs the melodic transcription, using the pYIN algorithm [7] to extract the fundamental frequency from the audio signal. The pYIN algorithm is a modification of the smoothing procedure of the YIN technique [1], introducing a probabilistic variant that outputs multiple pitch candidates along with the associated probabilities. It also employs a Hidden Markov Model (HMM) based on [16] to perform the pitch tracking, providing an improvement in the accuracy of the standard YIN. The extracted frame-wise sequence f_0 is then segmented and labeled into segments of music notes using the hysteresis approach of [11]. After, in this stage, we introduced a new alignment procedure, where the transcribed sung segments are aligned with the music score. This procedure converts group of melodic segments into atomic unities (music notes) and allows a direct comparison (note against note) between the transcribed melody and the ground truth. In the second stage, a probabilistic classifier performs the note-based evaluation. The algorithm takes the generated sequence of notes from the previous stage and applies a Bayesian classifier to evaluate the accuracy of the parameters pitch, onset and offset. At this stage, a rejection procedure is also introduced to map the doubt from the categorization (correct or incorrect) given by the expert listeners. These stages are described next.

4.1 Melodic alignment

To evaluate the solfège performance, a comparison of the singing performance with the target music score (ground truth) is required. Thus, after obtaining the automatic melodic transcription (using [7] and [11]), it is still necessary to connect each transcribed note with its corresponding note in the music score.

The first challenge is the fact that the melodic transcription often generates groups of fragmented notes (segments), which should be mapped to only one element of the ground truth. Each melodic fragment is represented by f_{il} , where i is the segment index and l is the relative index of each frame within this segment. Additionally, as in [10], there is no assumption of synchronization by a metronome in our approach, so that the transcribed notes might be misaligned. In [10], an integrated dynamic time warping procedure (DTW) was employed to perform the time alignment in a frame-wise fashion. However, in some cases, the boundary condition of the DTW algorithm might propagate the accumulated matching error, which causes an undesirable alignment between the transcribed sequence and the ground truth.

Here, we propose a new alignment process that, at the same time, groups note fragments and also maps the resulting block with the correspondent music note in the ground truth. Despite being similar to the DTW approach, it does not propagate the cumulative error since it does not need to obey the boundary condition of the DTW algorithm. The joint grouping/alignment process was designed as a brute force algorithm that is implemented using a cost matrix C . For each note k in the ground truth, the algorithm computes the cumulative distance measure considering all possibilities of grouping of adjacent segments, starting at segment index i and stopping at segment index j . This algorithm is efficiently built with the support of a 3D data structure, as depicted in Figure 1a. Thus, for each possible combination (k, i, j) , a dissimilarity measure is computed as

$$C(k, i, j) = \alpha_1 \Delta f(k, i, j) + \alpha_2 \Delta d(k, i, j) + \alpha_3 \Delta s(k, i, j) + \alpha_4 \Delta e(k, i, j), \quad (1)$$

where

$$\Delta f = |f_k^{gt} - \text{median}(f_{i,1} \dots f_{j,l_{max}})| \quad (2)$$

is the pitch distance between the ground truth note k and the median values of f_0 belonging to the range starting at first frame of the segment i and finishing at the last frame l_{max} of the segment j ,

$$\Delta d = |D_k^{gt} - \sum_{m=i}^j D_m| \quad (3)$$

measures the duration difference (in seconds) between the note k in the ground truth (D_k^{gt}) and the group formed from segment i to j in the transcribed melody (D_i is the duration of segment i),

$$\Delta s = |S_k^{gt} - S_i| \quad (4)$$

accounts for the delay or advance (in seconds) of the onset of the first segment of the selected group and the ground

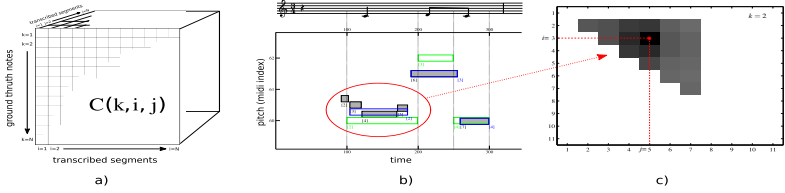


Figure 1: (a) 3D structure used to compute the similarities between the transcribed melodic segments and the music score. (b) Grouping process of several segments (gray) into one music note (blue). (c) The best grouping for the note k in the ground truth is found by the indexes i (first element) and j (last element), which minimize the function $C(k, i, j)$.

truth note k , and analogously

$$\Delta e = |E_k^{gt} - E_j| \quad (5)$$

accounts for the delay or advance of the offset (in seconds). The coefficients α_i are weights to balance the individual contribution of each measure, and our experiments show that $\alpha_1 = 1.0$, $\alpha_2 = 2.0$, $\alpha_3 = 2.0$, $\alpha_4 = 2.0$ is a good combination.

The grouping process and its mapping to the ground truth sequence is achieved by a function

$$v(k) = (i_k, j_k) = \underset{i,j}{\operatorname{argmin}} C(k, i, j), \quad (6)$$

so that each note k is mapped to the group of segments from indices i (first segment) to j (last segment), obtaining the final and consolidated transcribed note.

The computational complexity of the alignment process in the worst case is $\mathcal{O}(MN^2)$, where M is the number of music notes in the ground truth and N is the number of melodic segments. However, the inclusion of components Δs and Δe in Eq. (1) makes the magnitude of the dissimilarity measure to grow fast when the group of segments is far from the expected time position. As a consequence, it is possible to interrupt the brute force search loop in a few iterations by limiting the value of $C(k, i, j)$. Furthermore, the window of evaluation containing the melodic segments can be restricted to begin closer to the target note. This process will also decrease the computational cost and also avoid eventual local minimum issues in Eq. (6). Figure 1b illustrates one example of the grouping and alignment process, in which six segments are mapped into three notes.

4.2 Note-based evaluation

After the alignment achieved by the melodic transcription, the system performs the note-based assessment. Distinct probability density functions are modeled to represent the correct and incorrect sung notes, regarding individually to the pitch (Δf , in midi scale), onset (Δs , in seconds) and offset (Δe , in seconds) deviations. For each sung note, a Bayesian classifier assigns the parameters pitch, onset and offset into correct φ or incorrect $\bar{\varphi}$ categories. Next, the Bayesian classification process will be explained, focusing on the Δf (pitch) parameter. However, it is worth noting

that the classification process is also individually applied to Δs and Δe in an analogous way.

Figure 2a shows the histograms of the pitch deviations for correct and incorrect categories based on the expert's evaluation, denoted by $\varphi_{\Delta f}$ and $\bar{\varphi}_{\Delta f}$, respectively. As it can be observed, the histogram of $\varphi_{\Delta f}$ presents a sharp peak close the origin (related to low pitch errors), as expected. Nevertheless, the two categories present considerable overlap, corroborating the discrepancies in the accuracy evaluation by experts when for intermediate errors in the pitch. In fact, since we had used the individual ratings of each note from all evaluators to build the histograms, the pitch deviation Δf related to a note that received conflicting labels among the evaluators contributes both for the histograms of $\varphi_{\Delta f}$ and $\bar{\varphi}_{\Delta f}$.

A conditional probability density function is then estimated from the distributions of Δf for each class $r \in \{\varphi_{\Delta f}, \bar{\varphi}_{\Delta f}\}$, so that a posterior probability (that can be considered a measure of confidence) can be easily obtained. Among several existing parametric probability density functions (PDFs) for modeling positive random variables, the Gamma distribution was chosen because it has been successfully used to model similar problems [18], which have similar characteristics to our data, such as single mode and frequently skewed shape. The *gamma* PDF, parameterized by the two positive parameters shape α_r and scale θ_r , is given by:

$$p(\Delta f|r) \sim Ga(\Delta f; \alpha_r, \theta_r) = \frac{\Delta f^{\alpha_r-1} e^{-\frac{\Delta f}{\theta_r}}}{\Gamma(\alpha_r) \theta_r^{\alpha_r}}, \quad (7)$$

where Γ is the *gamma* function.

The shape (α_r) and scale (θ_r) parameters for each class $r \in \{\varphi, \bar{\varphi}\}$ were estimated using a maximum likelihood approach [15]. Given the PDFs $p(\Delta f|\varphi)$ and $p(\Delta f|\bar{\varphi})$, we can estimate the *posterior* probability of the pitch of a correct/incorrect sung note by using the Bayes rule [2]:

$$p(r|\Delta f) = \frac{p(\Delta f|r)P(r)}{p(\Delta f)}, \quad (8)$$

where $p(\Delta f) = p(\Delta f|\varphi)P(\varphi) + p(\Delta f|\bar{\varphi})P(\bar{\varphi})$ is the overall distribution of Δf , and the *prior* probabilities $P(\varphi)$ and $P(\bar{\varphi})$ are defined as equiprobable.

Figure 2b illustrates the decision boundary for $\varphi_{\Delta f}$ and $\bar{\varphi}_{\Delta f}$ as a red vertical dashed line, and it can be observed

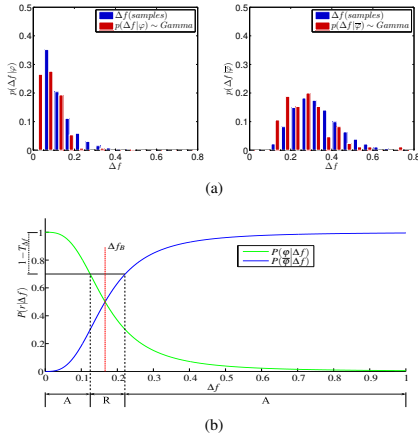


Figure 2: (a) Histogram of Δf for classes φ and $\bar{\varphi}$ along with fitted Gamma PDFs. (b) Posterior probabilities, along with acceptance and rejection regions.

that there is a “fuzzy” decision boundary around it. In this region, there is considerable overlap between $p(\Delta f|\varphi)$ and $p(\Delta f|\bar{\varphi})$, where the winning posterior probability to be just a little above 0.5. Since this overlap region is caused in part by conflicting labels from the evaluators, an appropriate option is to reject samples that fall inside this fuzzy region. As in [18], the errors (or misclassifications) are converted into rejections using the Bayes rejection rule for the minimum error [21]. The rejection rule splits the sample space into an acceptance region A and a rejection region R , that is given by:

$$R(T_{\Delta f}) = \{\Delta f | 1 - \max_r p(r|\Delta f) > T_{\Delta f}\}, \quad (9)$$

$$A(T_{\Delta f}) = \{\Delta f | 1 - \max_r p(r|\Delta f) \leq T_{\Delta f}\}, \quad (10)$$

where the threshold $T_{\Delta f}$ balances the tradeoff between the number of rejected samples and the error rate $e(T_{\Delta f})$, given by:

$$e(T_{\Delta f}) = \sum_{\Delta f \in A(T_{\Delta f})} \left(1 - \max_r p(r|\Delta f)\right) p(\Delta f). \quad (11)$$

The choice of the threshold $T_{\Delta f} = 0.33$ was determined from a set of experiments where the classification accuracy and the number of rejections were taken into account (more details about this choice are presented in section 5). The posterior probabilities and the boundaries between regions A and R generated by this threshold are shown in Figure 2b.

Thus, regarding the pitch accuracy and using the Bayesian classifier given by Eq. (8) in combination with the rejection procedure provided by Eqs. (9) and (10), each sung note is classified into three possible classes: correct, incorrect, or undetermined (reject). When a classification is

done (correct or incorrect), the corresponding probability measure is also used to provide a meaningful feedback of confidence to the user. The whole note-based evaluation process is also done independently for the onset and offset note accuracy. This means that, for each sung note, the system output gives individual class labels and confidence measures for pitch, onset and offset.

5. EXPERIMENTAL RESULTS

Aiming to extract an objective evaluation of the proposed solfège assessment system, a set of experiments were conducted using the annotated audio dataset described in Section 3. From the audio recordings, we extracted the melodic transcriptions, which were subsequently aligned with the ground truth, as described in Section 4.1. The pitch, onset and offset deviations (Δf , Δs and Δe) were computed from the comparison between the ground truth and the aligned melodies, and a subset of the samples was used to estimate the parameters required in the corresponding Gamma PDFs. The remaining samples were reserved to test the model.

For the validation scheme, we used a 10-fold cross-validation scheme, in which the dataset is split randomly into ten equal parts. For each round of the cross-validation, 9 folds are used to train the probabilistic model and the remaining fold is used to validate the Bayesian classifier described in Section 4.2. In our experiments, we used the Bayesian classifier with and without the rejection rule. In both situations, the system classifies each parameter (pitch, onset, offset) of each sung note in two possible categories: correct or incorrect (when the rejection rule was applied, some notes were kept unclassified).

Table 1 shows the confusion matrices generated by the Bayesian classifiers for the pitch, onset and offset without the rejection rule, and the accuracy is over 90% for the three analyzed parameters. Also, the system tends to produce more false negatives (i.e., mark as incorrect a correctly sung note) than false positives, particularly for the offset parameters, being a “rigid” evaluator. The misclassification errors are caused by two main reasons: first, a possible bad melodic transcription and/or bad alignment between the sung fragments and the ground truth can introduce errors on the similarities measures; second, the disagreement between the human evaluators generated an inherently fuzzy region near to the decision boundary. In fact, as noted in Section 3, 10 to 15% of the notes presented strong disagreement among the evaluators, so that the ground truth label may not be reliable.

The rejection rule provided by Eq. 9 avoids the classification of samples that potentially fall inside this fuzzy region. The effect of varying the rejection thresholds in the percentage of accepted samples and also the accuracy for the pitch, onset and offset analysis is shown in Figure 3. As expected, lower thresholds decrease the number of accepted samples and increases the accuracy rate. Although the definition of an optimal value for the threshold is difficult, the accuracy should be as maximum as possible while the number of rejected samples should be minimal. As the focus of this work is on music education, we believe it is

		Output Class		Total Accuracy
		$\varphi_{\Delta f}$	$\bar{\varphi}_{\Delta f}$	
Target Class	$\varphi_{\Delta f}$	88.99%	11.01%	90.86%
	$\bar{\varphi}_{\Delta f}$	7.27%	92.73%	

(a) Pitch evaluation

		Output Class		Total Accuracy
		$\varphi_{\Delta s}$	$\bar{\varphi}_{\Delta s}$	
Target Class	$\varphi_{\Delta s}$	89.17%	10.83%	90.22%
	$\bar{\varphi}_{\Delta s}$	8.74%	91.26%	

(b) Onset evaluation

		Output Class		Total Accuracy
		$\varphi_{\Delta e}$	$\bar{\varphi}_{\Delta e}$	
Target Class	$\varphi_{\Delta e}$	84.71%	15.29%	91.08%
	$\bar{\varphi}_{\Delta e}$	2.54%	97.46%	

(c) Offset evaluation

Table 1: Evaluation of the proposed approach using 10-Folds cross validation without the Bayesian rejection rule.

		Output Class		Total Accuracy
		$\varphi_{\Delta f}$	$\bar{\varphi}_{\Delta f}$	
Target Class	$\varphi_{\Delta f}$	94.45%	5.55%	95.96%
	$\bar{\varphi}_{\Delta f}$	2.54%	97.46%	

(a) Pitch evaluation: $T_{\Delta f} = 0.33$

		Output Class		Total Accuracy
		$\varphi_{\Delta s}$	$\bar{\varphi}_{\Delta s}$	
Target Class	$\varphi_{\Delta s}$	94.17%	5.83%	93.42%
	$\bar{\varphi}_{\Delta s}$	7.34%	92.66%	

(b) Onset evaluation: $T_{\Delta s} = 0.31$

		Output Class		Total Accuracy
		$\varphi_{\Delta e}$	$\bar{\varphi}_{\Delta e}$	
Target Class	$\varphi_{\Delta e}$	91.64%	8.36%	94.55%
	$\bar{\varphi}_{\Delta e}$	2.54%	97.46%	

(c) Offset evaluation: $T_{\Delta e} = 0.39$

Table 2: Evaluation of the proposed approach using 10-Folds cross validation with the Bayesian rejection rule. The system can answer in 90% of the times, increasing the final accuracy in almost 4%.

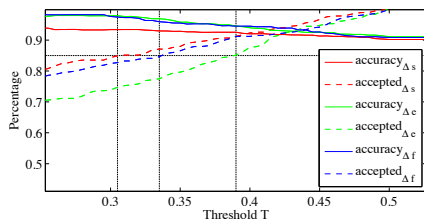


Figure 3: Comparative of the accuracy versus the number of non-rejected samples. Solid lines show the accuracy evolution, which are affected by the thresholds $T_{\Delta f}$ (pitch), $T_{\Delta s}$ (onset), and $T_{\Delta e}$ (offset).

preferred to not have an answer than to provide an incorrect feedback. Based on this assumption, and also considering that the percentage of samples with doubt from the expert evaluation is over 10%, we decided to set all thresholds to reject 15% of the samples in average.

Table 2 shows the accuracy evaluation for the 10-fold experiment using the Bayesian classifier with the rejection rule, in which the rejection thresholds $T_{\Delta f}$ (pitch), $T_{\Delta s}$ (onset) and $T_{\Delta e}$ (offset) were set so that 15% of the samples are rejected, matching approximately the percentage of samples with dubious labels. As it can be observed, the overall accuracies for all analyzed parameters increased in 3-5% when compared to the option without rejection, reaching up to almost 96% accuracy. Also, the number of false negatives was greatly reduced, particularly for the offset evaluation. This fact indicates that when in doubt, the evaluators tend to label a note as correct rather than incorrect. Furthermore, 32-35% of the rejected samples received 3 agreeing votes by the experts, which means that our system is removing more than twice of the samples related to the experts' doubt when compared with the whole dataset.

6. CONCLUSION

This paper presented a note-by-note approach for automatic solfège assessment focused on musical education, in which each sung note is evaluated considering the human evaluation perception in small scale, focused on the parameters of pitch, onset and offset at a specific part of the solfège practice. The proposed system uses melodic transcription techniques to extract the sung notes from the audio signal, and the sequence of melodic segments is subsequently processed by a two stage algorithm. In the first stage, an aggregation process was introduced to perform the temporal alignment between the transcribed melody and the music score (ground truth). This stage implicitly aggregates and links the best combination of the extracted melodic segments with the expected notes in the ground truth. The proposed alignment process does not impose the DTW boundary condition between the two sequences, avoiding the propagation of the accumulated matching error. In the second stage, a Bayesian classifier is used to evaluate the accuracy of each detected sung note. This statistical model was trained using a combination of the extracted measures (Δf , Δs , and Δe) with the individual scores provided by a committee of expert listeners.

Experimental results indicate that the classification scheme achieved accuracy rates in the range 90-91% without using the rejection rule (i.e., feedback for all evaluated notes), and 93-96% using the Bayesian rejection procedure (for the chosen thresholds, our tool is able to give feedback in 85% of the trials in average). Besides the classification label (correct, incorrect or undefined), the system also provides probability measure, which helps to indicate how likely correct or incorrect was the performance of the sung note. As future work, new research is planned to integrate new audio features, as well as the usage of lyrics analysis, to improve the segmentation and alignment on the first stage of this approach.

Acknowledgements: Thanks to CAPES Foundation, Ministry of Education of Brazil, scholarship BEX-2106/13-2.

7. REFERENCES

- [1] Alain de Cheveigné and Hideki Kawahara. YIN, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.
- [2] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, pages 24–27, 2001.
- [3] Emilia Gómez and J Bonada. Towards computer-assisted flamenco transcription: An experimental comparison of automatic transcription algorithms as applied to a cappella singing. *Computer Music Journal*, 37:73–90, 2013.
- [4] Mayank Vibhuti Jha and Preeti Rao. Assessing vowel quality for singing evaluation. In *Proceedings of the National Conference on Communications (NCC)*, pages 1–5, Kharagpur, India, Feb 2012.
- [5] Anssi Klapuri and Manuel Davy. *Signal Processing Methods for Music Transcription*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, pages 361–390, 2006.
- [6] Chang-Hung Lin, Yuan-Shan Lee, Ming-Yen Chen, and Jia-Ching Wang. Automatic singing evaluating system based on acoustic features and rhythm. In *Proceedings of the IEEE International Conference on Orange Technologies (ICOT), 2014*, pages 165–168, Sept 2014.
- [7] Matthias Mauch and Simon Dixon. pyin: A fundamental frequency estimator using probabilistic threshold distributions. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*, pages 659–663, May 2014.
- [8] Sai Sumanth Miryala, Ranjita Bhagwan, Monojit Choudhury, and Kalika Bali. Automatically identifying vocal expressions for music transcription. In *Proceedings of the 14th International Society of Music Information Retrieval, ISMIR 2013, Curitiba, Brazil, November 4–8*, pages 239–244, 2013.
- [9] Emilio Molina, Ana M. Barbancho, Lorenzo J. Tardón, and Isabel Barbancho. Evaluation framework for automatic singing transcription. In *Proceedings of the 15th International Society for Music Information Retrieval Conference, ISMIR 2014, Taipei, Taiwan, October 27–31*, pages 567–572, 2014.
- [10] Emilio Molina, Isabel Barbancho, Emilia Gómez, Ana Maria Barbancho, and Lorenzo J. Tardón. Fundamental frequency alignment vs. note-based melodic similarity for singing voice assessment. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP, 2013)*, pages 744–748, May 2013.
- [11] Emilio Molina, Lorenzo J. Tardón, Ana M. Barbancho, and Isabel Barbancho. Siph: Singing transcription based on hysteresis defined on the pitch-time curve. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(2):252–263, Feb 2015.
- [12] Meinard Müller. *Information Retrieval for Music and Motion*. Springer Verlag, pages 69–74, 2007.
- [13] Dorothy Payne. Essential skills, part 1 of 4: Essential skills for promoting a lifelong love of music and music making. *American Music Teacher*, February–March 2005.
- [14] Graham E. Poliner, Daniel P. W. Ellis, A. F. Ehmann, Emilia Gómez, S. Streich, and Beesuan Ong. Melody transcription from music audio: Approaches and evaluation. *IEEE Transactions on Audio, Speech and Language Processing*, 15(4):1247–1256, 2007.
- [15] Kandethody M. Ramachandran and Cris P. Tsokos. *Mathematical Statistics with Applications*. Elsevier Academic Press, 2009.
- [16] Matti Ryynänen. Probabilistic modelling of note events in the transcription of monophonic melodies. Master’s thesis, Tampere University of Technology, March 2004.
- [17] Matti Ryynänen and Anssi Klapuri. Modelling of note events for singing transcription. In *Proc. ISCA Tutorial and Research Workshop on Statistical and Perceptual Audio*, Jeju, Korea, October 2004.
- [18] Rodrigo Schramm, Cláudio Rosito Jung, and Eduardo Reck Miranda. Dynamic time warping for music conducting gestures evaluation. *Multimedia, IEEE Transactions on*, 17(2):243–255, Feb 2015.
- [19] Timo Viitaniemi, Anssi Klapuri, and Antti Eronen. A probabilistic model for the transcription of single-voice melodies. In *Tampere University of Technology*, pages 59–63, 2003.
- [20] Joel Wapnick and Elizabeth Ekholm. Expert consensus in solo voice performance evaluation. *Journal of Voice*, 11(4):429 – 436, 1997.
- [21] Andrew R. Webb. *Statistical Pattern Recognition*. Wiley, Chichester, UK, 3 edition, pages 8–17, 2011.

EVALUATING CONFLICT MANAGEMENT MECHANISMS FOR ONLINE SOCIAL JUKEBOXES

Felipe Vieira Nazareno Andrade

Department of Systems and Computing, Universidade Federal de Campina Grande, Brazil

{felipev, nazareno}@isd.ufcg.edu.br

ABSTRACT

Social music listening is a prevalent and often fruitful experience. Social jukeboxes are systems that enable social music listening with listeners collaboratively choosing the music to be played. Naturally, because music tastes are diverse, using social jukeboxes often involves conflicting interests. Because of that, virtually all social jukeboxes incorporate conflict management mechanisms. In contrast with their widespread use, however, little attention has been given to evaluating how different conflict management mechanisms function to preserve the positive experience of music listeners. This paper presents an experiment with three conflict management mechanisms and three groups of listeners. The mechanisms were chosen to represent those most commonly used in the state of the practice. Our study employs a mixed-methods approach to quantitatively analyze listeners' satisfaction and to examine their impressions and views on conflict, conflict management mechanisms, and social jukeboxing.

1. INTRODUCTION

The act of listening to music together is ubiquitous. In many situations, the choice of the music to be played for a group is done by an authority, such as a performer or a DJ; in other situations, groups rely on more democratic choices through social jukeboxes. Such devices have varying implementations in industry and have received attention from academia. In the latter, research has observed systems which arbitrate the selection of songs in gyms considering the musical tastes of those attending the gym [15], and systems that democratize the choice of music to be played in parties [10, 17], public spaces [16] and in cars [18]. In industry, Plug.DJ [3] (three million registered accounts), the recently shut down Soundrop [6] (peaked at nearly 49 thousand monthly active users) and the mobile applications Noisspot [1], PlayMySong [2], Rockbot [5], and Secret.DJ [7] (all of them with more than ten thousand downloads on virtual stores) are some commercial systems that presently have a significant user base.

Because people are often affected by the music heard in an environment [11], sharing the choice of music to be heard may lead to pleasant or dissatisfying experiences. In-

deed, in the presence of diverse musical tastes, it is likely that there will be conflicts in choosing music collectively. In the simplest case, one member of the group may like a genre or specific songs disliked by others. Even for participants that share similar tastes, one of them may be at a given moment interested in relaxing songs, while another participant is interested in increasing arousal. Tory et al. [10] and O'Hara et al. [16] have documented examples of such conflicts in the context of social jukeboxes.

To prevent that conflicts cause unpleasant experiences, it is central that social jukeboxes have mechanisms that manage such conflicts. Some of the aforementioned systems rely on voting to allow users to communicate their preferences. In part of these systems, this feedback also serves as an input to choose music based on the preference of the majority. However, in spite of the necessary and common use of conflict management mechanisms in social jukeboxes, there has been little or no comparative scientific evaluation of such mechanisms.

This work contributes to filling this gap by studying the use of three conflict management mechanisms in the same social jukeboxing system. The three mechanisms studied are present in multiple solutions in the state of the practice of social jukeboxes, and aim to represent significant points in the design space of conflict management mechanisms. Experiments were conducted with three user groups, each using the social jukebox in their natural settings. Our evaluation uses a mixed methods approach combining quantitative measures of user satisfaction and textured impressions stemming from semi-structured interviews in combination with observation reports and chat logs.

By analysing user satisfaction data, our results confirm that in spite of conceptual differences, the three conflict management mechanisms provide a significant gain in user satisfaction when compared to a baseline social jukebox with no mechanism. Moreover, the up/downvoting mechanism provides the highest satisfaction among the mechanisms we experiment with. A qualitative analysis of interviews, observation notes, and chat logs suggests that the effectiveness of voting is related to its interaction demands and the feedback it provides. Furthermore, analysing such data highlights other forms of conflicts and opportunities for the design of new conflict management mechanisms.

2. ONLINE SOCIAL JUKEBOXES AND CONFLICT

Akin to the jukebox metaphor, in online social jukeboxes users add songs to a queue to be played. This choice of



© Felipe Vieira, Nazareno Andrade. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Felipe Vieira, Nazareno Andrade. "Evaluating conflict management mechanisms for online social jukeboxes", 16th International Society for Music Information Retrieval Conference, 2015.

songs is the primary source of conflict, as users may disagree on the best song to be played in a given moment. Examining the industry and social jukeboxes in the research literature, we identify three mechanisms most often used to manage conflict: *like/dislike* feedback, *up/down voting* of songs in a queue, and a *skip* feature. Like/dislike is present in all systems mentioned except Jukola, up/down voting is used in Soundrop, Noispot, Rockbot and Jukola, and skip is implemented in Plug.DJ, Noispot and Jukola.

These three conflict management mechanisms can be easily evidenced in the observed jukeboxes. Like/dislike feedback is comprised of messages from users about the song currently playing. As such, it does not directly or immediately affect the music playing; in the presence of conflict it only conveys to the person responsible for the song the desire that future choices are different. This is the less intrusive mechanism. Up/down voting, in turn, allows for the group to change the order of songs that will be played next. If users downvote a song, this both communicates their negative preference and delays the song start. This delaying represents a more intrusive approach to manage conflicts by avoiding songs that will not satisfy some participants. Finally, skipping gives the group means to directly interfere in a song that is presently playing.

Allowing users to express their appreciation for some content is a widespread feature in social media. Cheng et al. [13] have found that in large-scale systems, this type of mechanism can lead to significant changes in the author's future behaviour by attaching more quality to the content shared after negative feedback. The mechanism of affecting the next song to be played by up/down voting on the queue items is perhaps the most straightforward mechanism of democratizing music choice. It also resembles approaches applied in different settings such as social Q&A or media aggregating sites such as Reddit [4], where users are able to choose which shared content is going to be most evident in the website by up/down voting posts. The possibility of abruptly stopping a song execution through skip seems to be more specific of social music systems, but has been recognized as valuable to avoid mood-breaking songs [18] and to prevent frequent users from the frustration of hearing the same song multiple times [16].

It is worthwhile mentioning that although there are a number of conflict management mechanisms used in social jukebox systems, to the best of our knowledge there has been no experimental study that compares the effectiveness of conflict management mechanisms for these systems.

3. THREE CHOSEN MECHANISMS IN AN ONLINE JUKEBOX

Given the state of the practice observed in conflict management for social jukeboxes, we opted to experiment with the three mechanisms identified as most often employed: like/dislike feedback, up/down voting and skip. These mechanisms were implemented in a social jukebox developed by the authors and named WePlay, which has its basic interface shown in Figure 1.

WePlay allows for a group of users to synchronously

listen to music coming from a shared queue of songs to which all can contribute. Each user can contribute as many songs as desired by searching these songs on YouTube and adding to a queue visible by all. The queue lists songs, but not the users who contributed the songs. Besides features available to the users, WePlay also allows an experimenter to alternate the conflict management mechanism exposed to users at will. The implementation of the three mechanisms is detailed next.

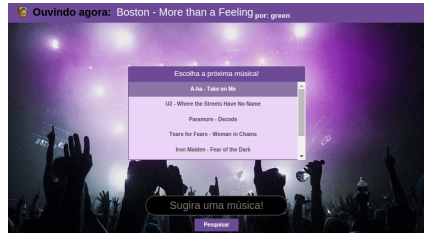


Figure 1. The interface of WePlay, the social jukebox system used in our experiments

3.1 Like/Dislike

Similarly to prevalent mechanisms in online social media, when this mechanism is available, users have access to like and dislike buttons next to the name of the song presently playing, as shown in Figure 2. Similar to the social jukebox systems we observed, this explicit feedback does not directly control which song will play next. Instead, it serves as a message to the user who queued the song stating how welcome that song has been considered by current listeners. In WePlay, only one immutable feedback may be provided per song. Moreover, the number of likes and dislikes is visible for all listeners, but no listener has access to the list of users who liked or disliked a song. Finally, when this mechanism is enabled, users are able to see a list of previously played songs and the feedback they received.

3.2 Up/down voting

By using the up/down voting mechanism in WePlay, users can vote up or down songs in the queue. Users can cast one vote per song, also immutable. After each vote, songs are ordered according to their balance, calculated as the difference between its positive and negative votes. The queue interface is depicted in Figure 3. Neither voters nor current balance are shown in the interface, but the highest-ranked song is always highlighted. In the event of a tie, the timestamp is considered the tiebreaker, awarding highest rank to the song first suggested to the system.

3.3 Skip

This mechanism allows the jukebox users to collectively skip the current song. If enough users manifest such will,



Figure 2. The like/dislike mechanism and the feedback history



Figure 3. The up/downvoting mechanism

the song is then immediately skipped and the next song from the queue starts playing. To manifest opinions about the song, users can cast positive or negative votes about it. Considering the number of listeners n , the number of positive votes p , and the negative votes s , if the overall satisfaction $o = ((p-s)/n) + 1$ of the current song reaches a value below the threshold of 0.5, the song is skipped, following the skip mechanism idealized heuristically by the original authors of a side project which was adapted to result in our WePlay and maintained due to the similarity of the original use of the system and our experimental scenario.

4. EXPERIMENTAL SETUP

Three different groups were recruited to participate in our experiments. Recruitment was done using the social network of authors, primarily targeting groups of potential users of an online social jukebox that would be available over multiple consecutive days for the experiments. Participants from the first two groups are undergraduate students, graduate students or researchers working in the same university as the authors, totaling 18 participants (16 males and 2 females, average age 25.2). Participants in these groups are work colleagues who used the system during normal workdays, and were collocated in the same or adjacent rooms during the experiment. The third group is comprised by friends of one of the authors (10 males, average age 25) who have known each other for years, work in diverse fields, used the system during leisure time, and were located in different places in a common city. In all three groups, our goal is to study the use of the social jukebox system integrated in the subjects' routine, aiming at the ecological validity in social listening research suggested

by North [9].

Each experiment lasted for the period of five days. All participants were submitted to a briefing explaining how the system would work and describing the experiment dynamics (e.g. what conflict management mechanisms would be enabled on each day). None of the participants was aware of the specific details of the research. All participants were informed that the goal of the experiment was to evaluate multiple designs of the social jukebox, and agreed to use the system for the duration of the experiment, and to have data collected during this time to be used in the research. In the week after each experiment, users were interviewed about their experience using semi-structured interviews. Four, seven and seven participants were interviewed respectively on groups one, two and three, totalling 18 interviews. Interviews lasted on average 15 minutes.

During the experiment, each group first interacted with the social jukebox using no conflict management mechanism in a situation we dub baseline. After the baseline, the other conflict resolution mechanisms were available one at a time and in the same order for all groups, for one complete day each. On the fifth day, all three mechanisms were available for participants, in a setting we call combined mechanisms. During the complete experiment, participants and the experimenter shared a text chat room using Google Hangouts. This communication channel was meant primarily for the experimenter to answer questions, but also hosted diverse conversations among participants during the experiments.

The social jukebox used in the experiments is instrumented to provide detailed usage information through logs. Furthermore, to gauge participant's overall satisfaction with the system, the jukebox asked participants to provide every 30 minutes their level of satisfaction through a 5-point likert scale in a form which asked users to explicitly state their *satisfaction with recently played songs*. Although the action of listening to music is often a background task and users could forget to answer this request, whenever a new request was made, participants were reminded to answer the from through the group chat.

5. CONFLICT SITUATIONS

As expected, the interviews and our observation of system usage revealed conflict situations. Overall, our data shows some conflicts related to a participant having an aversion to a song proposed by another participant. Such aversion may be related to one's musical identity [8] (*Everytime she suggested I immediately voted negative, because of her musical taste*¹) and were perceived to affect satisfaction (*There was a moment when I felt upset about the songs. They were putting some songs like funk, and I don't really like funk. But it was a radio, and it was in a democracy style, so I had to listen to that. Or in some moments I was very dissatisfied. There were some songs I cannot stand... Some musical styles.*).

¹ Quotes from the interviews are presented henceforth in italics and parenthesis. All quotes were translated from Portuguese to English by the authors.

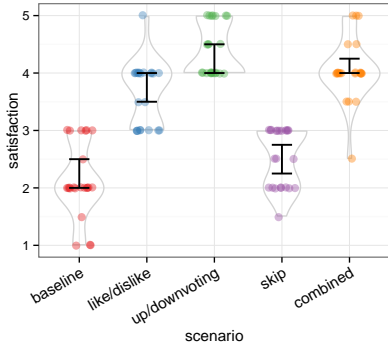


Figure 4. Distribution of median satisfaction reported by users in each of the scenarios. The violin glyphs encode density. Error bars represent 95% confidence intervals for the medians. In comparing the intervals, one should take into account that samples are paired; this pairing results in, up/down voting having a significantly higher median satisfaction than the combined scenario, and in skip significantly outperforming the baseline.

A second and minor source of conflict relevant to the mechanisms we experimented with is related to gaming the mechanisms and trolling. Participants reported their tendency to game the voting mechanism, and trolling behavior by users.

6. CONFLICT MANAGEMENT AND USER SATISFACTION

Our quantitative data contains multiple satisfaction ratings for each participant in each of the five different designs: baseline, like/dislike, up/downvoting, skip, and all mechanisms. In the following, each participant’s satisfaction is summarized as the median of the ratings provided in each design.

Albeit conceptually categorical, likert scales data in the form employed in our experiment can be reliably used in numerical statistical tests [14, 19]. A normality test however points that the satisfaction data is not normal, (Shapiro-Wilk test, $p < .01$ for all five scenarios). This observation combined with the sample size ($N < 30$ for all samples) leads us to use non-parametric tests to compare participant satisfactions.

Participants’ satisfaction and the 95% confidence interval for the median satisfaction across participants are shown in Figure 4. It is readily apparent that like/dislike, up/downvoting, and the combine mechanisms all lead to significantly higher user satisfaction than the baseline system. A rank-sum comparison using Mann-Whitney paired one-tailed tests reveals that all mechanisms provide significantly higher satisfaction than the baseline ($p < .02$

for all designs. Like/dislike: $V = 276$, up/downvoting: $V = 276$, skip: $V = 117.5$).

Comparing the mechanisms among themselves, we see that up/down voting has at the same time the highest median satisfaction and the smallest dispersion in satisfaction values. The overall higher satisfaction of participants when using the voting mechanism is also confirmed by a rank-sum comparison with the combined scenario (Mann-Whitney paired one-tailed, $p = .02$, $V = 75$). Since the participants in group 3 had different backgrounds to those in the other two groups, the previous statistical tests were repeated withholding data pertaining to group 3. The results of this test have similar outcomes.

Next to up/down voting, the combination of mechanisms resulted in the second highest satisfaction scores. This may reflect the availability of the high-performing up/down voting mechanism in the combination. The second best performing sole mechanism is like/dislike. The mechanism that provided the smallest increase in satisfaction in our experiments was skipping.

Finally, Figure 5 compares satisfactions reported by participants in the different groups. The general pattern is the same for all three groups. This is so in spite of the relatively different context in which group 3 used the system.

Together, our quantitative results suggest that the best strategy for a designer considering implementing a conflict management mechanism in a social jukebox system is to focus on up/down voting. In the next section we elaborate on the reasons behind participants’ preferences, and on other relevant episodes in the experiments.

7. IMPRESSIONS ABOUT CONFLICT MANAGEMENT

Besides the quantitative data, we now turn to analyze collected interviews, observations taken by the experimenters, and chat history among participants. The qualitative data was explored using Grounded Theory [12] methods for coding and categorizing quotes, and to analyse the emergent themes.

7.1 Mechanisms’ effectiveness

An overall positive effect of the conflict management mechanisms reported by users is the possibility of communicating of one’s identity and preferences to negotiate a common ground and reduce conflict (... *and I found it very interesting the little window on the bottom of the screen where we could see our latest ratings. It’s useful when you’re choosing your next song and you don’t want to pick a song nobody likes*).

Focusing on up/down voting, this mechanism seems to offer a particularly convenient trade-off between expressing preferences on multiple songs and having to often interrupt other tasks to use the social jukebox (... *and I also thought the songs list [with the up/ down votes] very interesting because we could express our opinions and go back to our main activities, avoiding to open the system all the time, focusing on our jobs and still making our voices*

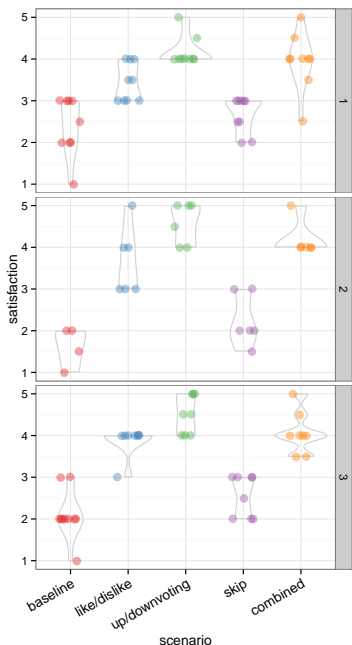


Figure 5. Distribution of median satisfaction reported by users in each of the scenarios, with users divided in the three experimental groups.

active inside the system). Both like/dislike and skip must be performed on a song while it was playing, and thus required more frequent interaction with the system (*I used [the like/dislike feature] on almost all songs, except when I was really busy with work*).

In our experiments, positive feedback was seen as more usual, and negative feedback as related to more extreme cases (*I only used the dislike feature when a song was a really really bad choice or if the other guy was clearly trolling*), or to constantly send explicit messages to users with mismatched musical tastes about the incoherence of their choices (*There's no significant difference between my positive and negative voting, I guess, except that when [a participant's name] suggested. Then I always voted negative due to her musical taste*).

The skip feature as implemented in our experiments had a major limitation related to presence. Our system accounted for listeners as active if they are logged in, and demanded that a proportion of active listeners voted for skipping to actually skip the song. Because music listening was a background activity, and participants interleaved this activity with attention to other tasks or even being temporar-

ily physically away from the computer, there were often insufficient votes for skipping (*It was hard to see the skip feature happening. It barely happened, and in a rare moment when [the song] was skipped I think it only happened because that song had a really big rejection or Although I think the skip is a great idea it almost didn't happen, and when it happened I thought it was because our room wasn't full yet and a few negative feedbacks were enough to skip the song, which sadly came to be a song of mine...*).

7.2 Gaming and trolling

Gaming conflict resolution and trolling are two often reported phenomena in online communities. In our experiments, both behaviours happened and were commented on during interviews. For example, one participant reported strategies for imposing their choices on the group: *There were several times when I tried to downvote all songs except mine's, so I could just upvote any of my songs and place them at the top, playing it before the other's choices. It didn't succeed because the guys discovered my strategy and started to downvote my songs. I tried also to dislike all the other's songs, hoping that the system skipped those, but that didn't happen*.

On a different occasion, because participants were mostly friends, there were participants who posted nonsense songs or repeatedly posted a song related to some meme as a joke with the group. Although subverting the rules and joking may reinforce social ties, in our experiments it had detrimental effects (*There was a time when we had a song related to a viral, and because of that the song got repeated over and over again, so as I couldn't handle it I took my earphone off and put it on a little later to hear some new songs, if that was the case*). Another user clearly stated he was motivated by jokingly annoying others (*It was my fuel. When it annoyed people, I'd put the song again*).

7.3 Design Suggestions

After being exposed to four situations in conflict management, participants were also asked about their views about the design space of social jukeboxes.

A participant suggested that more mechanisms to communicate musical identity may be of use, and that perhaps allowing one to specify such identity explicitly could contribute to reduce conflict by enabling semi-automatic song choice (*I think a good way [to increase conflict management] is to allow user profiling, something like: an user has three musical preferences, so when he starts using the system he could be asked to fill a form stating those three choices, and after that the system could check who is online and select the next song according to the intersection of musical tastes*).

Further room to increase the convenience of expressing preferences in the system when music is a background task was also mentioned. A participant suggested the use of smartphones for enabling interaction in such cases (*It would be great if we have a tool to facilitate the voting process, because we can only vote at the web page, and*

sometimes we are [on our desk but] not using our PC but we keep listening the songs, so if we could, for example, vote in a song using our smartphone that could make the democratic process even better).

A more challenging suggestion to experiment with that was mentioned by multiple participants is the possibility of punishing users perceived as trolls (*It came to a point when I had enough of [another participant's name]'s songs. I really wish he was unable to suggest songs, so the system could at least enable the chance of banning a song which received too much negative votes, but actually I think it would be even greater if we could "mute" a specific user, removing the access to the features and only allowing him to listen the songs suggested by the others).*

8. IMPLICATIONS FOR DESIGN

Our experiment evaluates three commonly used conflict management mechanisms in online social jukeboxes. Together, the quantitative and qualitative results point for multiple implications for designers of social jukeboxes.

In our experience, conflicts were relatively easy to reproduce. Participants of all of our experiments were already friends or colleagues, and had multiple communication channels besides the social jukebox. Yet, conflicts related to incompatibility in music tastes and different intentions in listening to music on a given moment were reported to influence satisfaction with the music listening experience.

Quantitatively, all three conflict management mechanisms led to significant improvements in user satisfaction with the music played in the experiments. It is also notable that the mechanisms provided such increase in the presence of conversations both face-to-face and through online chat to manage the same conflicts. This result suggests that simple mechanisms effectively complement more textured social interactions to negotiate this type of conflict.

Comparing mechanisms, our results point that up/down voting songs on the queue leads to the highest overall user satisfaction. Both the median and minimum satisfaction of participants were the highest with this mechanism. A qualitative analysis points that up and downvoting seems to be on a sweet spot of the design space as it allows for conveniently sparse batch interactions with the system, combined with an informative log of past song evaluations. These results, together with the ease of implementing up and down voting recommends that present designers consider this mechanism. Moreover, it suggests that conflict resolution mechanisms for background music listening take into account the frequency of interaction with the system.

With respect to the log of past evaluations, our analysis suggests it has a constructive role in preventing conflicts. Our experiment does not allow for isolating its effect, but suggests this and other mechanisms that allow users or the group to express their taste are likely to contribute to conflict management. Indeed, this direction is similar to the common behaviour of stating group norms explicitly in many online communities.

Other relevant aspects that arose in our analyses were the limited effect of the skip mechanisms and the presence of gaming and trolling. The former is chiefly related to difficulties in detecting and communicating user presence while music was a background task. As a result, the system perceived too many users as active, and participants felt that voting for skipping was not effective. Detecting which users are presently interacting with the system and devising a skipping policy more easily understandable may lead to different results, and our mechanism allow limited conclusions in this perspective.

With respect to gaming and trolling, our experiments highlight that these phenomena happen in social music listening even for small-scale scenarios. From our observations, the mechanisms we experimented with were robust to gaming. Trolling in our setting was related to jokes from a user that reduced the satisfaction of others – which nevertheless were reported as trolling in the interviews. The interviews suggest that mechanisms to regulate such behaviours may contribute to the success of online social jukeboxes.

9. LIMITATIONS AND FUTURE WORK

This work contributes preliminary findings to an understanding of the effectiveness of multiple points in the design space of conflict management mechanisms for online social jukeboxes. In doing so, it has a number of limitations and leaves open questions for future work.

An issue that markedly limits the generalizability of our findings is related to the characteristics of our sample of users. All participants were already acquainted, and by and large male. Replicating our experiment with more groups with different compositions is a direct and necessary extension of this work. This is necessary to examine the degree to which the context, closeness, and size of the group affect our results. Moreover, understanding whether and how direct conversation interferes with conflict management also seems like a promising avenue of research.

Another point that demands further study is the analysis of other policies for each of the mechanisms examined here. Other policies for consolidating votes, skip requests, and like and dislike feedback may be more suitable for certain contexts. Also, experimenting with other policies for skipping seems particularly relevant, given the feedback from the participants in our experiments.

Finally, our experience highlights and commends for future work the benefits of conducting similar research in a naturalistic setting. Observing participants use the system in their normal routine, and participating in social listening with colleagues and friends helped unveil a number of relevant observations in our research.

10. ACKNOWLEDGEMENTS

We would like to thank the developers of Rádio LSD, on which WePlay was based – particularly Abmar Barros – and the participants of our experiment.

11. REFERENCES

- [1] Noisspot website. <http://noisspot.com/>. Accessed: 2015-04-23.
- [2] Playmysong website. <http://www.playmysong.com/>. Accessed: 2015-04-23.
- [3] Plug.dj website. <https://plug.dj/>. Accessed: 2015-04-23.
- [4] Reddit website. <http://www.reddit.com/>. Accessed: 2015-04-23.
- [5] Rockbot website. <https://rockbot.com/>. Accessed: 2015-04-23.
- [6] Soundrop website. <http://soundrop.fm/>. Accessed: 2015-04-23.
- [7] Soundrop website. <http://www.secretdj.com/>. Accessed: 2015-04-23.
- [8] J. Hargreaves A. North, D. Hargreaves. The functions of music in everyday life: Redefining the social in music psychology. In *Psychology of Music*, pages 84–95, 1999.
- [9] J. Hargreaves A. North, D. Hargreaves. Uses of music in everyday lifes. In *Music Perception*, pages 41–77., 2004.
- [10] M. Tory. D. Sprague, F. Wu. Music selection using the partyvote democratic jukebox. In *Proceedings of AVI 2008*, 2008.
- [11] T. DeNora. *Music and everyday life*. Cambridge: Cambridge University Press., 2000.
- [12] Monique Hennink, Inge Hutter, and Ajay Bailey. *Qualitative research methods*. Sage, 2010.
- [13] J. Leskovec J. Cheng, C. Danescu-Niculescu-Mizil. How community feedback shapes user behavior. In *ICWSM*, 2014.
- [14] D. Dodou J. de Winter. Five-point likert items: t test versus mann-whitney-wilcoxon. In *Pract. Assess. Res. Eval.* 15 (11), 1–12, 2010.
- [15] T. Anagnost J. McCarthy. Musicfx: An arbiter of group preferences for computer supported collaborative workouts. In *Proceedings of the ACM 1998 Conf. on Comp. Support. Coop. Work (CSCW 98)*, pages 363–372, 1998.
- [16] M. Jansen A. Unger H. Jeffries P. Macer K. O’Hara, M. Lipson. Jukola: Democratic music choice in a public space. In *Proceedings of DIS 2004*, Boston, MA, 2004.
- [17] D. Nichols S. Cunningham. Exploring social music behavior: An investigation of music selection at parties. In *Proceedings of ISMIR 2009*, pages 747–752., 2009.
- [18] B. Bainbridge H. Ali S. Jo Cunningham, D. M. Nichols. Social music in cars. In *Proceedings of 15th International Society for Music Information Retrieval Conference*, 2014.
- [19] C. Wu. An empirical study on the transformation of likert-scale data to numerical scores. In *Applied Mathematical Sciences, Vol. 1 No. 58*, 2851–2861, 2007.

PARTICLE FILTERS FOR EFFICIENT METER TRACKING WITH DYNAMIC BAYESIAN NETWORKS

Ajay Srinivasamurthy*

ajays.murthy@upf.edu

Ali Taylan Cemgil†

taylan.cemgil@boun.edu.tr

Andre Holzapfel†

andre@rhythmos.org

Xavier Serra*

xavier.serra@upf.edu

*Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain

†Dept. of Computer Engineering, Boğaziçi University, Istanbul, Turkey

ABSTRACT

Recent approaches in meter tracking have successfully applied Bayesian models. While the proposed models can be adapted to different musical styles, the applicability of these flexible methods so far is limited because the application of exact inference is computationally demanding. More efficient approximate inference algorithms using particle filters (PF) can be developed to overcome this limitation. In this paper, we assume that the type of meter of a piece is known, and use this knowledge to simplify an existing Bayesian model with the goal of incorporating a more diverse observation model. We then propose Particle Filter based inference schemes for both the original model and the simplification. We compare the results obtained from exact and approximate inference in terms of meter tracking accuracy as well as in terms of computational demands. Evaluations are performed using corpora of Carnatic music from India and a collection of Ballroom dances. We document that the approximate methods perform similar to exact inference, at a lower computational cost. Furthermore, we show that the inference schemes remain accurate for long and full length recordings in Carnatic music.

1. INTRODUCTION

Rhythm analysis of musical audio signals plays an important role in Music Information Retrieval (MIR) research. Many of the works in MIR related to rhythm attempt to establish a relation between the audio signal and the underlying musical meter. For instance, in the task of beat tracking, the goal is to obtain an alignment of the metrical level referred to as the *tactus* [15] to an audio signal, see [8] for a list of references to recent beat tracking algorithms. Tracking meter at a higher metrical level is a task pursued under the title of downbeat detection. Approaches were presented that either attempt to identify the downbeat separately from the *tactus* [7], or that pursue beat tracking and downbeat detection as a combined task [11, 17]. The combined task of beat and downbeat detection is what we refer to as meter tracking, since it aims at aligning several

levels of a known meter to an audio recording of a music performance.

Many applications can profit from accurate meter or beat tracking. Some synchronization tasks, such as the one presented in [6], tracking the beat is sufficient. However, other applications, such as musical structure analysis [16] can profit from a more detailed understanding of the temporal structure of a performance. Approaches that can achieve such an analysis for a wider variety of music usually incorporate machine learning strategies to adapt to new styles. For instance, Böck et al. [1] presented a method for beat tracking in various styles that achieves high accuracy using recurrent neural networks that were adapted to the individual styles. The task of downbeat tracking was addressed in [4] using a set of deep belief networks trained on various features, and the regularity of the outputs was enforced by incorporating a simple hidden Markov model (HMM). The task of meter tracking was combined with the determination of the type of meter in [9], using a Dynamic Bayesian Network (DBN) similar to the one applied in [1].

A significant shortcoming of the mentioned tracking approaches is that their flexibility in terms of musical style comes at an increased computational cost, either in terms of time spent for the training of networks [1, 4], or in terms of long inference times [9]. In the present paper, we approach faster inference in a DBN in two ways. Firstly, we propose a change to the model structure as presented in [9, 14] that enables faster inference by simplifying the independence assumptions between the variables of the model. The proposed simplification also addresses one of the main limiting factors in most of the approaches so far: a simplistic observation model that cannot effectively handle diversity in rhythmic patterns. Secondly, one reason for long inference times of the model proposed in [9] is the utilization of exact inference in an HMM, which discretizes the hidden variables of the state space to compute the most likely path in the exact posterior distribution using the Viterbi algorithm. Here, we avoid the discretization of the state space by approximating the posterior using particle filter methods [3]. The biggest challenge in applying such approximate methods to meter tracking is the multi-modality of the underlying posterior distribution [22] due to the ambiguity inherent to musical meter. Recently, methods were proposed that overcome these challenges [14]. We outline the existing [9, 14] and the proposed simplified model, and compare the performance of exact and approximate inference schemes for both the models, in terms of meter track-



© Ajay Srinivasamurthy, Andre Holzapfel, Ali Taylan Cemgil, Xavier Serra.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Ajay Srinivasamurthy, Andre Holzapfel, Ali Taylan Cemgil, Xavier Serra. "Particle Filters for Efficient Meter Tracking with Dynamic Bayesian Networks", 16th International Society for Music Information Retrieval Conference, 2015.

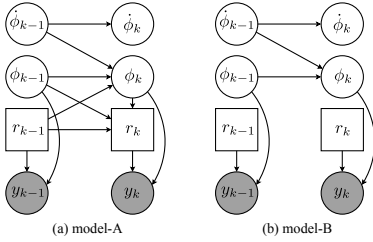


Figure 1: The DBNs used in this paper: circles and squares denote continuous and discrete variables, respectively. Gray nodes and white nodes represent observed and latent variables, respectively. Model-A is from [14] and model-B is the proposed simplification.

ing accuracy and computational demands.

Carnatic music, the art music tradition from South India is a representative case to study in this context. Meter in Carnatic music is defined by the *tāla*, which are time cycles with three metrical levels: the *sama* (downbeat, the first pulse of the cycle), beat, and the subdivision level (a comprehensive account on Carnatic music is provided in [19]). In performances of Carnatic music, however, large degrees of freedom are taken by the musicians to conceal the underlying meter and to add metrical ambiguity, for instance by changing the beat structure during a metrical cycle. This playful rhythmic character of Carnatic music leads to our hypothesis that meter tracking should be able to profit from a diverse observation model. Most of the rhythmic structures, melodic phrases, and stratal elements are tightly associated with the cycles of the *tāla* [20] and hence tracking the *sama* (downbeat) is an important MIR task in Carnatic music, which is the main focus of this paper. We will also evaluate if meter tracking in Carnatic music can profit from including a richer observation model that can incorporate information from multiple patterns.

In order to further illustrate the ability of the approach to generalize, it will be additionally evaluated on a corpus of Ballroom dances [5]. Furthermore, reproducibility will be ensured by providing free access for research purposes to all code repositories and datasets¹. We begin by describing the models and inference schemes that we use for meter tracking.

2. MODEL STRUCTURE

We compare two different Bayesian models for the task of meter tracking. The first model (model-A), depicted in Figure 1a, is identical to the model used in [9, 14] and was initially proposed in [24]. We propose and discuss a simplification to model-A for the task of meter tracking, shown as model-B in Figure 1b. Model-B uses a diverse observation model and can be applied if the type of meter is known in advance. It is to be noted that model-A can also be used for inferring the type of meter, though we apply it in this paper only for meter tracking.

¹Please see the companion webpage for more details: <http://compmusic.upf.edu/ismir-2015-pf>

In a DBN, an observed sequence of features derived from an audio signal $\mathbf{y}_{1:K} = \{\mathbf{y}_1, \dots, \mathbf{y}_K\}$ is generated by a sequence of hidden (unknown) variables $\mathbf{x}_{1:K} = \{\mathbf{x}_1, \dots, \mathbf{x}_K\}$, where K is the length of the sequence (number of audio frames in an audio excerpt). The joint probability distribution of hidden and observed variables factorizes as,

$$P(\mathbf{y}_{1:K}; \mathbf{x}_{0:K}) = P(\mathbf{x}_0) \cdot \prod_{k=1}^K P(\mathbf{x}_k | \mathbf{x}_{k-1}) P(\mathbf{y}_k | \mathbf{x}_k) \quad (1)$$

where, $P(\mathbf{x}_0)$ is the initial state distribution, $P(\mathbf{x}_k | \mathbf{x}_{k-1})$ is the transition model, and $P(\mathbf{y}_k | \mathbf{x}_k)$ is the observation model.

2.1 Hidden Variables

At each audio frame k , the hidden variables describe the state of a hypothetical bar pointer $\mathbf{x}_k = [\phi_k \ r_k]$, representing the bar position, instantaneous tempo and a rhythmic pattern indicator, respectively (see Figure 1 of [23] for an illustration).

- *Bar position:* The bar position $\phi \in [0, M)$, where M is the length of the bar (cycle). The maximum value of M depends on the longest bar (cycle) that is tracked. We set the length of a full note to 1600, and scale other bar (cycle) lengths accordingly.
- *Rhythmic pattern:* The rhythmic pattern variable $r \in \{1, \dots, R\}$ is an indicator variable to select one of the R observation models corresponding to each bar (cycle) length rhythmic pattern learned from data. Each pattern has a bar length M and a number of beats B , which are assumed to be known in advance, i.e. the goal is the tracking of a known metrical structure.
- *Instantaneous tempo:* Instantaneous tempo $\dot{\phi}$ is the rate at which the bar position variable progresses through the cycle at each time frame, measured in bar positions per time frame. The range of the variable $\dot{\phi}_k \in [\dot{\phi}_{\min}, \dot{\phi}_{\max}]$ depends on the length of the cycle M and the hop size ($\Delta = 0.02s$ used in this paper), and can be preset or learned from data. A tempo value of $\dot{\phi}_k$ corresponds to a bar (cycle) length of $(\Delta \cdot M \cdot \dot{\phi}_k)$ seconds and $(60 \cdot B \cdot \dot{\phi}_k \cdot (M \cdot \Delta))$ beats per minute.

The conditional dependence relations between the variables for both the models are shown in Figure 1.

2.2 Initial state distribution

We can use $P(\mathbf{x}_0)$ to incorporate prior information about the metrical structure of the music into the model. In this paper, we assume uniform priors on all variables, within the allowed ranges of tempo.

2.3 Model-A: Transition and Observation model

Due to the conditional dependence relations in Figure 1a, the transition model factorizes as,

$$P(\mathbf{x}_k | \mathbf{x}_{k-1}) = P(\phi_k | \phi_{k-1}, \dot{\phi}_{k-1}, r_{k-1}) P(\dot{\phi}_k | \dot{\phi}_{k-1}) \times P(r_k | r_{k-1}, \phi_k, \phi_{k-1}) \quad (2)$$

Each of the terms in Eqn (2) are defined in Eqns (3)–(5).

$$P(\phi_k | \phi_{k-1}, \dot{\phi}_{k-1}, r_{k-1}) = \mathbb{1}_{\phi} \quad (3)$$

where $\mathbb{1}_\phi$ is an indicator function that takes a value of one if $\phi_k = (\phi_{k-1} + \dot{\phi}_{k-1}) \bmod(M(r_k))$ and zero otherwise (in our case, $M(r_k) = M$), meaning that the bar position advances at the rate of the instantaneous tempo variable, and folds back when it crosses the maximum value that is defined by the length M of the metrical cycle.

$$P(\dot{\phi}_k | \dot{\phi}_{k-1}) \propto \mathcal{N}(\dot{\phi}_{k-1}, \sigma_\phi^2) \times \mathbb{1}_\phi \quad (4)$$

where $\mathbb{1}_\phi$ is an indicator function that equals one if $\dot{\phi}_k \in [\dot{\phi}_{\min}, \dot{\phi}_{\max}]$ and zero otherwise. $\mathcal{N}(\mu, \sigma)$ denotes a normal distribution with mean μ and standard deviation σ .

$$P(r_k | r_{k-1}, \phi_k, \phi_{k-1}) = \begin{cases} \mathbf{A}(r_{k-1}, r_k) & \text{if } \phi_k < \phi_{k-1} \\ \mathbb{1}_r & \text{else} \end{cases} \quad (5)$$

where, $\mathbf{A}(i, j)$ is the time-homogeneous transition probability from r_i to r_j , and $\mathbb{1}_r$ is an indicator function that equals one when $r_k = r_{k-1}$ and zero otherwise. Since the rhythmic patterns are one bar (cycle) in length, pattern transitions are allowed only at the end of the bar (cycle). The pattern transition probabilities are learned from data.

The observation model is identical to the one used in [14], and depends only on the bar position and rhythmic pattern variables. We use a two dimensional spectral flux feature in two frequency bands (Low: ≤ 250 Hz, High: > 250 Hz). Using beat and downbeat annotated training data, a k-means clustering algorithm clusters and assigns each bar of the dataset (represented by a point in a 128-dimensional space) to one of the R rhythmic patterns. We then discretize the bar into 64^{th} note cells (corresponding to 25 bar positions with $M_{\max} = 1600$), collect all the features within the cell for each pattern, and compute the maximum likelihood estimates of the parameters of a two component Gaussian Mixture Model (GMM). The observation probability hence is computed as,

$$P(\mathbf{y} | \mathbf{x}) = P(\mathbf{y} | \phi, r) = \sum_{i=1}^2 w_{\phi, r, i} \mathcal{N}(\mathbf{y}; \boldsymbol{\mu}_{\phi, r, i}, \boldsymbol{\Sigma}_{\phi, r, i}) \quad (6)$$

where, $\mathcal{N}(\mathbf{y}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ denotes a normal distribution and for the mixture component i , $w_{\phi, r, i}$, $\boldsymbol{\mu}_{\phi, r, i}$ and $\boldsymbol{\Sigma}_{\phi, r, i}$ are the component weight, mean (2-dimensional) and the covariance matrix (2×2), respectively.

2.4 Model-B: Transition and Observation model

We propose a simpler model-B (Figure 1b) that uses a diverse mixture observation model incorporating observations from multiple rhythmic patterns. Since all the rhythmic patterns belong to the same type of meter (tāla), we can simplify model-A to track only the ϕ and $\dot{\phi}$ variables while using an observation model that computes the likelihood of an observation by marginalizing over all the patterns. The motivation for this simplification is two-fold: the inference is simplified, and we can increase the influence of diverse patterns that occur throughout a metrical cycle in the inference.

For model-B, we first define $\mathbf{x}_k = [\boldsymbol{\alpha}_k, r_k]$, where $\boldsymbol{\alpha}_k = [\phi_k, \dot{\phi}_k]$. Based on the conditional dependence relations in Figure 1b, the transition model now is,

$$P(\mathbf{x}_k | \mathbf{x}_{k-1}) = P(\boldsymbol{\alpha}_k | \boldsymbol{\alpha}_{k-1}) = P(\phi_k | \phi_{k-1}, \dot{\phi}_{k-1}) P(\dot{\phi}_k | \dot{\phi}_{k-1}) \quad (7)$$

Eqns. (3) and (4) remain identical apart from the removal of the dependence on r_{k-1} in Eqn (3). The observation model is a pre-computed mixture observation model computed from Eqn (6) by marginalizing over the patterns, assuming equal priors.

$$P(\mathbf{y} | \boldsymbol{\alpha}) \propto \prod_{j=1}^R P(\mathbf{y} | \phi, r = j) \quad (8)$$

3. INFERENCE METHODS

The goal of inference is to find a hidden variable sequence that maximizes the posterior probability of the hidden states given an observed sequence of features: a maximum *a posteriori* (MAP) sequence $\mathbf{x}_{1:K}^*$ that maximizes $P(\mathbf{x}_{1:K} | \mathbf{y}_{1:K})$. The inferred hidden variable sequence $\mathbf{x}_{1:K}^*$ can then be translated into a sequence downbeat (sama) instants ($\phi_k^* = 0$), beat instants ($\phi_k^* = i \cdot M/B$, $i = 1, \dots, B$), and the local instantaneous tempo ($\dot{\phi}_k^*$). We describe two different inference schemes, an exact inference using an HMM in a discretized state space, and an approximate inference using particle filters using the continuous values of ϕ and $\dot{\phi}$.

3.1 Hidden Markov model (HMM)

By discretizing the continuous variables bar position and tempo, we can perform an exact inference using HMM. We use the discretization proposed in [14], by replacing the continuous variables ϕ and $\dot{\phi}$ by their discretized counterparts, $m \in \{1, 2, \dots, [M]\}$ and $n \in \{n_{\min}, n_{\min} + 1, \dots, n_{\max}\}$, with the discrete tempo limits as $n_{\min} = \lfloor \dot{\phi}_{\min} \rfloor$ and $N = n_{\max} = \lceil \dot{\phi}_{\max} \rceil$, where $\lceil \cdot \rceil$ and $\lfloor \cdot \rfloor$ denote the ceil and floor operations, respectively. Eqns (2), (3) and (5) remain valid. We define the tempo transition probability within the allowed tempo range as,

$$P(n_k | n_{k-1}) = \begin{cases} 1 - p_n & \text{if } n_k = n_{k-1} \\ \frac{p_n}{2} & \text{if } n_k = n_{k-1} \pm 1 \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where p_n is the probability of tempo change. We use Viterbi algorithm [18] to obtain a MAP sequence of states with the HMM. We refer to the HMMs for inference from model-A and model-B as HMM_a and HMM_b, respectively.

The drawback of this approach is that the discretization has to be on a very fine grid in order to guarantee good performance, which leads to a prohibitively large state space and, as a consequence, to a computationally demanding inference. The size of the state space is $S = M \cdot N \cdot R$ and needs an $S \times S$ sized transition matrix. As an example, dividing a bar into $M = 1600$ position states, with $N = 15$ tempo states and $R = 4$ patterns, the size of the state space is $S = 96000$ states. The computational complexity of the Viterbi algorithm is $O(K \cdot |S|^2)$. Even though the state transition matrix is sparse due to lesser number of allowed transitions leading to a complexity of $O(K \cdot M \cdot R)$, the inference with HMM can become computationally prohibitive and does not scale well with increasing number of states. This problem can be overcome, for instance, by using approximate inference methods such as particle filters.

3.2 Particle Filter (PF)

Particle filters (or Sequential Monte Carlo methods) are a class of approximate inference algorithms to estimate the

posterior density of a state space. They overcome two main problems of the HMM: discretization of the state space and the quadratic scaling up of the size of state space with more number of variables. In addition, they can incorporate long term relationships between hidden variables.

The exact computation of the posterior $P(\mathbf{x}_{1:K}|\mathbf{y}_{1:K})$ is often intractable, but it can be evaluated pointwise. In particle filters, the posterior is approximated using a weighted set of points (known as particles) in the state space as,

$$P(\mathbf{x}_{1:K}|\mathbf{y}_{1:K}) \approx \sum_{i=1}^{N_p} w_K^{(i)} \delta(\mathbf{x}_{1:K} - \mathbf{x}_{1:K}^{(i)}) \quad (10)$$

Here, $\{\mathbf{x}_{1:K}^{(i)}\}$ is a set of points (particles) with associated weights $\{w_K^{(i)}\}$, $i = 1, \dots, N_p$, and $\mathbf{x}_{1:K}$ is the set of all state trajectories until frame K , while $\delta(x)$ is the Dirac delta function, $\delta(x) = 1$ if $x = 0$ and 0 otherwise. N_p is the number of particles.

To approximate the posterior pointwise, we need a suitable method to draw samples $\mathbf{x}_k^{(i)}$ and compute appropriate weights $w_k^{(i)}$ recursively at each time step. A simple approach is Sequential Importance Sampling (SIS) [3], where we sample from a *proposal* distribution $Q(\mathbf{x}_{1:K}|\mathbf{y}_{1:K})$ that has the same support and is as similar to the true (target) distribution $P(\mathbf{x}_{1:K}|\mathbf{y}_{1:K})$ as possible. To account for the fact that we sampled from a proposal and not the target, we attach an importance weight $w_K^{(i)}$ to each particle, computed as,

$$w_K^{(i)} = \frac{P(\mathbf{x}_{1:K}|\mathbf{y}_{1:K})}{Q(\mathbf{x}_{1:K}|\mathbf{y}_{1:K})} \quad (11)$$

With a suitable proposal density, these weights can be computed recursively as,

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{P(\mathbf{y}_k|\mathbf{x}_k^{(i)})P(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})}{Q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k)} \quad (12)$$

Following [14], we choose to sample from the transition probability $Q(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)}, \mathbf{y}_k) = P(\mathbf{x}_k^{(i)}|\mathbf{x}_{k-1}^{(i)})$, which reduces Eqn (12) to

$$w_k^{(i)} \propto w_{k-1}^{(i)} P(\mathbf{y}_k|\mathbf{x}_k^{(i)}) \quad (13)$$

The SIS algorithm derives samples by first sampling from proposal, in this case the transition probability and then computes weights according to Eqn (13). Once we determine the particle trajectories $\{\mathbf{x}_{1:K}^{(i)}\}$, we then select the trajectory $\mathbf{x}_{1:K}^{(i^*)}$ with the highest weight $w_K^{(i^*)}$ as the MAP state sequence.

Many extensions have been proposed to the basic SIS filter (see [3] for a comprehensive overview) to address several problems with it. We briefly mention some of the relevant extensions, emphasizing their key aspects. A more detailed description of the algorithms has been presented in [14]. The most challenging problem in particle filtering is the degeneracy problem, where within a short time, most of the particles have a weight close to zero, representing unlikely regions of state space. This is contrary to the ideal case when we want the proposal to match well with the target distribution leading to a uniform weight distribution with low variance. To reduce the variance of the particle weights, resampling steps are necessary, which replaces low weight particles with higher weight particles by

selecting particles with a probability proportional to their weights. Several resampling methods have been proposed, but we use systematic resampling in this paper as recommended in [3]. With resampling as the essential difference, the SIS filter with resampling is called as Sequential Importance Sampling/Resampling (SISR) filter.

In meter tracking, due to metrical ambiguities, the posterior distribution $P(\mathbf{x}_k|\mathbf{y}_{1:k})$ is highly multimodal. Resampling tends to lead to a concentration of particles in one mode of the posterior, while the remaining modes are not covered. One way to alleviate this problem is to compress the weights $w_k = w_k^{(i)}$, $i = 1, \dots, N_p$ by a monotonically increasing function to increase the weights of particles in low probability regions so that they can survive resampling. After resampling, the weights have to be unnormalized to give a valid probability distribution. This can be formulated as an Auxiliary Particle Filter (APF) [10]. Further, a system that is capable of handling metrical ambiguities must maintain this multimodality and be able to track several hypotheses together, which SISR and APF cannot do explicitly. A system called the Mixture Particle Filter (MPF) was proposed to track multiple hypotheses in [22], and was adapted to meter inference in [14].

In an MPF, each particle is assigned to a cluster that (ideally) represents a mode of the posterior. During resampling, the particles of a cluster interact only with particles of the same cluster. Resampling is done independently in each cluster, while maintaining the probability distribution intact. This way, all the modes of the posterior can be tracked through the whole audio piece, and the best hypothesis can be chosen at the end. We use an identical clustering scheme using a cyclic distance measure as described in [14] to track several different possible metrical positions at a given time. In the MPF, after an initial cluster assignment, we perform a re-clustering before every resampling step, merging or splitting clusters based on the average distance between cluster centroids. The clustering, merging and splitting of clusters is necessary to control the number of clusters, which ideally represents the number of modes in the posterior. The mixture particle filter can be combined with the Auxiliary resampling to give the Auxiliary Mixture Particle Filter (AMPF). As recommended in [14], we resample at a fixed interval T_s . It was shown in [14] that AMPF can be effectively used for the task of meter inference and tracking.

With model-A, we setup an AMPF (AMPF_a) to compute the pointwise estimates of the posterior of $\mathbf{x}_{1:K}$, represented by $\{w_{\mathbf{x}_k}^{(i)}, \mathbf{x}_{1:K}^{(i)}, i = 1, \dots, N_p\}$, where N_p is the number of particles and $w_{\mathbf{x}_k}^{(i)}$ are the weights corresponding to the particle trajectories $\mathbf{x}_{1:K}^{(i)}$. The weights are updated as in Eqn (13), using the observation model in Eqn (6). This particle filter is identical to the AMPF described in [14], however, in this paper it is evaluated for the first time assuming several patterns with transitions allowed.

For the simplified model-B, we setup AMPF_b similarly for $\alpha_{1:K}$, represented by $\{w_{\alpha_k}^{(i)}, \alpha_{1:K}^{(i)}, i = 1, \dots, N_p\}$, where $w_{\alpha_k}^{(i)}$ are the weights corresponding to the particle trajectories $\alpha_{1:K}^{(i)}$. Similar to Eqn (13), the weight updates

Algorithm 1 Outline of the AMPF_b algorithm

-
- 1: **for** $i = 1$ to N_p **do**
 - 2: Sample $(\alpha_0^{(i)}) \sim P(\phi_0)P(\dot{\phi}_0)$, set $w_{\alpha,0}^{(i)} = 1$ N_p
 - 3: Cluster $\{\phi_0^{(i)}\}$ and obtain cluster assignments $\{c_0^{(i)}\}$
 - 4: **for** $k = 1$ to K **do**
 - 5: **for** $i = 1$ to N_p **do**
 - 6: Sample $\phi_k^{(i)} \sim P(\phi_k^{(i)}|\phi_{k-1}^{(i)})$, Set $c_k^{(i)} = c_{k-1}^{(i)}$
 - 7: $\hat{w}_{\alpha,k}^{(i)} = w_{\alpha,k}^{(i)} \times \sum_{j=1}^R P(\mathbf{y}_k|\phi_k^{(i)}, r = j)$
 - 8: **for** $i = 1$ to N_p **do** ▷ Normalize weights
 - 9: $w_{\alpha,k}^{(i)} = \frac{\hat{w}_{\alpha,k}^{(i)}}{\sum_{i=1}^{N_p} \hat{w}_{\alpha,k}^{(i)}}$
 - 10: **if** $\text{mod}(k, T_s) = 0$ **then**
 - 11: Recluster and Resample $\{\alpha_k, w_{\alpha,k}\}$ and obtain $\{\hat{\alpha}_k, \hat{w}_{\alpha,k}\}$, update $\{c_k^{(i)}\}$
 - 12: **for** $i = 1$ to N_p **do**
 - 13: Set $\alpha_k^{(i)} = \hat{\alpha}_k^{(i)}$, $w_{\alpha,k} = \hat{w}_{\alpha,k}$
 - 14: Sample $\dot{\phi}_k^{(i)} \sim P(\dot{\phi}_k^{(i)}|\phi_{k-1}^{(i)})$
-

for AMPF_b are,

$$w_{\alpha,k}^{(i)} \propto w_{\alpha,k-1}^{(i)} P(\mathbf{y}_k|\alpha_k^{(i)}) \quad (14)$$

where $P(\mathbf{y}_k|\alpha_k^{(i)})$ is computed as in Eqn (8) by marginalizing $P(\mathbf{y}_k|\mathbf{x}_k^{(i)})$ over $r^{(i)}$. The AMPF_b enables therefore to incorporate the full expressivity of the observed patterns into the inference. An outline of AMPF_b is provided in Algorithm 1.

The complexity of the PF schemes scale linearly with N_p irrespective of the size of state space, leading to an efficient inference in large state spaces. Further, compared to the HMM using Viterbi decoding that has a space complexity of $O(K \cdot |S|)$, the PF needs to store just N_p state trajectories and weights, significantly reducing the memory requirements. An additional advantage is that the number of particles can be chosen based on the computational power we can afford, and we can make the state space larger with no or only a marginal increase in the computational requirements. Since the observation likelihood can be precomputed, inference with model-B requires much lower computational resources, with only a marginal increase in cost during inference with increase in number of patterns.

4. EXPERIMENTS

The experiments aim to compare the performance of the particle filter and the HMM inference schemes for meter tracking with both model-A and model-B. Further, we wish to see if using a larger number of patterns per rhythm class (tāla) improves meter tracking performance. Meter tracking is done for each type of meter (tāla) separately, in a two fold cross validation experiment.

4.1 Music Corpora

The primary dataset we evaluate on is the Carnatic music dataset (CMD) used in [9]. It includes 118 two minute long excerpts spanning four commonly used tālas as shown in Table 1, with a total duration of 236 minutes and over 5500

Tāla	M	B	#Excerpts CMD	#Pieces CMD _f
Ādi (8/8)	1600	8	30 (60)	50 (252.8)
Rūpaka (3/4)	1200	3	30 (60)	50 (267.4)
Mīśra chāpu (7/8)	1400	7	30 (60)	48 (342.1)
Khaṇḍa chāpu (5/8)	1000	5	28 (56)	28 (134.6)

Table 1: The Carnatic music datasets, showing the cycle length M used in the paper and the number of beats B for each tāla. The analogous time signature is also shown. CMD is a subset of CMD_f, with two minute excerpts from full pieces. The number of pieces/excerpts in both datasets is also shown, the numbers in parentheses indicate the total duration of audio in minutes.

sama instances. To test if the results extend to full pieces, we use the super set of CMD consisting of longer and full length pieces (called CMD_f) as used in [21]. CMD_f comprises about 16.6 hours of audio with over 22600 sama instances. For comparability, we also present results on the Ballroom dataset [5], using the annotations from [12].

4.2 Parameter Selection and Learning

The tempo ranges were manually set for Carnatic music as $\phi \in [4, 15]$ (cycle lengths between 1.33 s and 8 s) and $\dot{\phi} \in [6, 32]$ (bar lengths between 0.75 s to 5.3 s) for the Ballroom dataset. With $M_{\max} = 1600$ (corresponds to ādi tāla with 8 beats/cycle), the length of cycle M and the number of beats B for each tāla is shown in Table 1. For Ballroom dataset, we used $M = 1600$ and $M = 1200$ for tracking time signatures 4/4 and 3/4, respectively. For the HMM, we use $p_n = 0.02$ as in [12], and for the AMPF, we use $\sigma_\phi = 10^{-4} \cdot M$. We explore the performance with $R = \{1, 2, 4\}$, with the number of particles set to $N_p = 1500 \cdot R$. The other AMPF parameters are identical to the values used in [14].

4.3 Evaluation Measures

A variety of measures for evaluating beat and downbeat tracking performance are available (see [2] for a detailed overview and descriptions of the metrics listed below²). We chose two metrics that are characterized by a set of diverse properties and are widely used in beat tracking evaluation. We describe it for beats, but the definitions extend to downbeats/samas as well, with the same tolerances. We use the prefix ‘s-’ and ‘b-’ to distinguish between the performance measures of sama and beat tracking, respectively. *Fmeas* (F-measure): The F-measure (a number between 0 and 1) is computed from correctly detected beats within a window of ± 70 ms as the harmonic mean of the precision (the ratio between the number of correctly detected beats and all detected beats) and recall (the ratio between the number of correctly detected beats and the total annotated beats).

AMLI (Allowed Metrical Levels with no continuity required): In the AMLI measure (a number between 0 and 1), beat sequences are considered as correct if the beats occur on the off-beat, or are double or half of the annotated tempo, allowing for metrical ambiguities. The value of this

² We used the code available at <http://code.soundsoftware.ac.uk/projects/beat-evaluation/> with default settings

Measure R	Sama tracking						Beat tracking					
	s-Fmeas			s-AMLt			b-Fmeas			b-AMLt		
	1	2	4	1	2	4	1	2	4	1	2	4
HMM _a	0.733	0.736	0.713	0.837	0.837	0.804	0.85	0.847	0.850	0.868	0.874	0.852
AMPF _a	0.708	0.697	0.704	0.827	0.809	0.822	0.846	0.833	0.843	0.872	0.874	0.862
HMM _b	0.726	0.735	0.736	0.830	0.862	0.867	0.844	0.849	0.837	0.864	0.893	0.900
AMPF _b	0.690	0.712	0.735	0.832	0.842	0.853	0.833	0.838	0.846	0.869	0.888	0.890
Klapuri [11]	0.175			0.181			0.657			0.650		

Table 2: Meter tracking performance on CMD. In addition, the performance of meter tracking with the algorithm proposed in [11] is also shown for reference.

Dataset Measure	CMD _f		Ballroom	
	s-Fmeas	b-Fmeas	s-Fmeas	b-Fmeas
HMM _a	0.727	0.834	0.806	0.929
AMPF _b	0.728	0.834	0.793	0.930

Table 3: F-measure for meter tracking on CMD_f and the Ballroom dataset, with $R = 4$. Values in each column are not statistically significantly different.

measure is then the ratio between the number of correctly estimated beats divided by the number of annotated beats.

4.4 Results and Discussion

We report the average Fmeas and AMLt values for all excerpts over all the tālas for the HMM and AMPF schemes in Table 2. The results for AMPF are the mean values over three experiments. We conducted evaluations using several other measures as well without any qualitative change in results. Therefore, experimental results are documented using these two measures. We use a three-way ANOVA with tāla, inference scheme, and R as factors to assess statistically significant differences (at 5% significance levels).

In general, we see that the beat tracking performance is similar across all the inference schemes and values of R , with the b-Fmeas and b-AMLt values being comparable. This shows that adding a diverse observation model and additional patterns does not add a significant change, showing that handling pattern diversity is not needed for beat tracking.

For sama tracking, we see that the AMPFs show statistically equivalent performance to the HMMs. The simpler AMPF_b performs as good or better than AMPF_a, with a lower computational complexity. Higher number of patterns ($R > 1$) do not show significant improvement in tracking performance, despite a richer observation model. This observation needs further exploration to verify if incorporating more patterns with the currently used features helps to improve sama tracking. Further, s-AMLt is significantly larger than s-Fmeas and shows that there is a potential for improvement in tracking the correct metrical level.

Though we report only consolidated set of results averaged over all the tālas, the tracking performance is significantly poorer for ādi tāla (e.g. s-Fmeas = 0.4, b-Fmeas = 0.632 with AMPF_b and $R = 4$), with superior (and statistically equivalent) results with other three tālas (e.g. s-Fmeas = 0.849, b-Fmeas = 0.92 with AMPF_b and $R = 4$). This is attributed to the long cycle durations and a large variety of patterns in ādi tāla, which shows a definite scope for improvement using higher number of patterns and better

observation models.

We extend the evaluation and report the performance of HMM_a and the proposed AMPF_b on CMD_f and Ballroom datasets (in an identical setting, assuming that the meter type is known) in Table 3. We see that the observations from CMD extend to these datasets too. We further see a similar performance between CMD and CMD_f, that shows that the AMPF generalizes to longer and full length pieces.

One of the main advantages of model-B over model-A is the lower computational cost. For meter tracking under the conditions described, all the inference schemes have faster than real time execution. Inference in model-B is faster than that in model-A: model-B speeds up inference by a factor of about 5 for HMM and 2.5 for AMPF (for $R = 4$ and ādi tāla). Even in the smaller state space with model-B, HMM_b has a higher memory requirement than AMPF_b, which shows the utility of PF inference schemes.

5. CONCLUSIONS

For the task of meter tracking, we presented a simplified Bayesian model that incorporates a richer observation model. We compared the performance of an exact inference using an HMM using a discrete approximation of the models, with an approximate inference using an AMPF on the exact model. The simplified model leads to faster inference and a similar performance as the full model, with the performance extending to full length pieces and generalizing to different music styles. However, the proposed way to enrich the observation model did not lead to significant differences in performance. This might be caused by the simplistic audio features, and improving signal representations appears as a necessary next step. In the future, we plan to explore approximate inference in improved models (such as [13] using an improved state space discretization and tempo transition model) that also use better observation models and can effectively utilize multiple rhythmic patterns. We also plan to extend meter tracking to Hindustani music, where long cycles (longer than a minute) exist and hence present additional challenges.

Acknowledgments

This work is supported by the European Research Council (grant number 267583) and a Marie Curie Intra-European Fellowship (grant number 328379). The authors also thank Florian Krebs and Sebastian Böck at Johannes Kepler University, Linz, Austria for providing access to their code repositories.

6. REFERENCES

- [1] S. Böck, F. Krebs, and G. Widmer. A multi-model approach to beat tracking considering heterogeneous music styles. In *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 602–607, Taipei, Taiwan, 2014.
- [2] M. Davies, N. Degara, and M. D. Plumbley. Evaluation methods for musical audio beat tracking algorithms. *Queen Mary University of London, Technical Report C4DM-09-06*, 2009.
- [3] A. Doucet and A. M. Johansen. A tutorial on particle filtering and smoothing: Fifteen years later. *Handbook of Nonlinear Filtering*, 2009.
- [4] S. Durand, J. P. Bello, B. David, and G. Richard. Downbeat tracking with multiple features and deep neural networks. In *Proc. of the 40th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Brisbane, Australia, 2015.
- [5] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech and Language Processing*, 14(5):1832–1844, 2006.
- [6] D. K. Grunberg, A. M. Batula, and Y. E. Kim. Towards the development of robot musical audition. In *Proc. of the Music, Mind, and Invention Workshop (MMI)*, New Jersey, USA, 2012.
- [7] J. A. Hockman, M. E. P. Davies, and I. Fujinaga. One in the Jungle: Downbeat Detection in Hardcore, Jungle, and Drum and Bass. In *Proc. of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 169–174, Porto, Portugal, October 2012.
- [8] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon. Selective Sampling for Beat Tracking Evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(9):2539–2548, November 2012.
- [9] A. Holzapfel, F. Krebs, and A. Srinivasamurthy. Tracking the “odd”: Meter inference in a culturally diverse music corpus. In *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 425–430, Taipei, Taiwan, 2014.
- [10] A. Johansen and A. Doucet. A note on auxiliary particle filters. *Statistics and Probability Letters*, 78(12):1498–1504, 2008.
- [11] A. P. Klapuri, A. J. Eronen, and J. T. Astola. Analysis of the Meter of Acoustic Musical Signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):342–355, 2006.
- [12] F. Krebs, S. Böck, and G. Widmer. Rhythmic pattern modeling for beat- and downbeat tracking in musical audio. In *Proc. of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, pages 227–232, 2013.
- [13] F. Krebs, S. Böck, and G. Widmer. An efficient state-space model for joint tempo and meter tracking. In *Proc. of the 16th International Society for Music Information Retrieval Conference (ISMIR)*, Málaga, Spain, October 2015.
- [14] F. Krebs, A. Holzapfel, A. T. Cemgil, and G. Widmer. Inferring metrical structure in music using particle filters. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 23(5):817–827, May 2015.
- [15] J. London. *Hearing in Time: Psychological Aspects of Musical Meter*. Oxford University Press, Oxford, 2004.
- [16] J. Paulus, M. Müller, and A. Klapuri. Audio-based music structure analysis. In *Proc. of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 625–636, Utrecht, Netherlands, August 2010.
- [17] G. Peeters and H. Papadopoulos. Simultaneous beat and downbeat-tracking using a probabilistic framework: Theory and large-scale evaluation. *IEEE Transactions on Audio, Speech and Language Processing*, 19(6):1754–1769, 2011.
- [18] L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proc. of the IEEE*, 77(2):257–286, February 1989.
- [19] P. Sambamoorthy. *South Indian Music Vol. I-VI*. The Indian Music Publishing House, 1998.
- [20] A. Srinivasamurthy, A. Holzapfel, and X. Serra. In Search of Automatic Rhythm Analysis Methods for Turkish and Indian Art Music. *Journal of New Music Research*, 43(1):97–117, 2014.
- [21] A. Srinivasamurthy and X. Serra. A supervised approach to hierarchical metrical cycle tracking from audio music recordings. In *Proc. of the 39th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5237–5241, Florence, Italy, May 2014.
- [22] J. Vermaak, A. Doucet, and P. Pérez. Maintaining multimodality through mixture tracking. In *Proc. of the 9th IEEE International Conference on Computer Vision*, pages 1110–1116, Nice, France, October 2003.
- [23] N. Whiteley, A. T. Cemgil, and S. Godsill. Bayesian modelling of temporal structure in musical audio. In *Proc. of the 7th International Society for Music Information Retrieval Conference (ISMIR)*, Victoria, 2006.
- [24] N. Whiteley, A. T. Cemgil, and S. Godsill. Sequential inference of rhythmic structure in musical audio. In *Proc. of the 33rd IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, volume 4, pages 1321–1325, Honolulu, USA, April 2007.

ANALYSIS OF THE EVOLUTION OF RESEARCH GROUPS AND TOPICS IN THE ISMIR CONFERENCE

Mohamed Sordo
Center for Computational Science
University of Miami
msordo@miami.edu

Mitsunori Oghihara
Dept. of Computer Science
University of Miami
ogihara@cs.miami.edu

Stefan Wuchty
Dept. of Computer Science
University of Miami
wuchty@cs.miami.edu

ABSTRACT

We present an analysis of the topics and research groups that participated in the ISMIR conference over the last 15 years, based on its proceedings. While we first investigate the topological changes of the co-authorship network as well as topics over time, we also identify groups of researchers, allowing us to investigate their evolution and topic dependence. Notably, we find that large groups last longer if they actively alter their membership. Furthermore, such groups tend to cover a wider selection of topics, suggesting that a change of members as well as of research topics increases their adaptability. In turn, smaller groups show the opposite behavior, persisting longer if their membership is altered minimally and focus on a smaller set of topics. Finally, by analyzing the effect of group size and lifespan on research impact, we observed that papers penned by medium sized and long lasting groups tend to have a citation advantage.

1. INTRODUCTION

Music Information Retrieval (MIR) is an interdisciplinary research field that integrates a wide variety of research areas, including audio signal processing, musicology, music psychology and cognition, information retrieval, and human-computer interfaces. The collection of papers published in the annual proceedings of the ISMIR conference provides a wealth of information enabling us to mine for knowledge such as the networks of researchers that contribute papers and corresponding topics. Specifically, such abundant data allows us to explore two main research questions. First we focus on topics in the field. Given the breadth of the expertise of the field and the high speed at which the digital technologies are developing, we investigate if popular topics can be transient. Second, we study the stability of research groups that emerge from the co-authorships of manuscripts, focusing on their sizes, diversity of topics and competitiveness. While various approaches for the exploration of knowledge in the ISMIR


paper collection exist, we considered a combination of network and text analysis.

Recently, the analysis of scientific endeavors by investigating author relationships and their manuscripts provided insights into innovation and idea creation processes [3], inter-dependencies between disciplines [2], or potential high-impact discoveries [5]. Utilizing proceedings of the mainstream conference we provide such a map of the status and temporal evolution of the MIR field. To the best of our knowledge, only two studies on the nature of the ISMIR proceedings [7, 10] have been presented recently. Grachten et al. [7] applied text mining techniques and non-negative matrix factorization to identify topics and study their evolution over time. Lee et al. [10] applied simple text statistics to detect topics in paper titles and abstracts. In our case, we present a much broader analysis of research topics, that we map to categories that were defined by the ISMIR community. While Lee et al. [10] also presented a few statistics to identify patterns of co-authorship we model the co-authorship as a complex network and study its topology. Yet, the main contribution of this paper is the identification of research groups and their evolution over time, and especially their time and topic dependencies. In the context of this paper, we do not use the definition of a research group in its traditional sense (e.g., a research institution). Rather, we define it as a topological group in a co-authorship network.

As for the organization of the paper we first describe the network of collaborations among authors in section 2. In section 3, we provide an analysis of the manuscripts text contents with a generative mixture model, allowing us to find temporal trends in the popularity of topics over the years. In section 4 we identify research groups in the co-authorship network and analyze their evolution throughout the lifetime of the conference, investigating their time and topic dependencies. Finally, we discuss our findings in section 5.

2. CO-AUTHORSHIP ANALYSIS

Utilizing all manuscripts in the proceedings of the ISMIR conference from 2000-2014 we observed that the mean number of authors per manuscript is growing over time (Table 1), confirming previous results [18]. Starting with the proceedings of the 2000 conference, we added new manuscripts that were published in a given year to a grow-

 © Mohamed Sordo, Mitsunori Oghihara, Stefan Wuchty. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Mohamed Sordo, Mitsunori Oghihara, Stefan Wuchty. "Analysis of the Evolution of Research Groups and Topics in the ISMIR Conference", 16th International Society for Music Information Retrieval Conference, 2015.

Year	Papers	Authors	Authors/Paper
2000	35	63	1.94
2001	37	82	2.54
2002	53	113	2.36
2003	47	108	2.74
2004	104	213	2.41
2005	114	232	2.73
2006	87	185	2.56
2007	127	267	2.84
2008	105	262	2.93
2009	123	292	3.05
2010	110	262	3.01
2011	133	320	2.97
2012	101	264	3.21
2013	98	232	3.02
2014	106	273	3.24

Table 1. For each year, we show the total number of papers and authors that published a manuscript in the proceedings of the ISMIR conference.

ing pool of papers. Based on such cumulative sets of manuscripts, we constructed undirected unweighted networks G , where nodes represent authors, while edges indicate their co-authorships up to a given year.

Table 2 suggests that the cumulative networks drastically increased in size over time, a statistics that coincides with an increasing number of collaboration partners (i.e. mean degree $\langle k \rangle$).

Another important measure of social networks is the clustering coefficient, reflecting the transitivity of a network. In particular, this network parameter determines the fraction of edges that appear between the neighbors of a given author over all such possible links [17]. Table 2 indicates that the co-authorship networks appear increasingly clustered, resembling a well known feature of other social networks from different domains [9, 12]. Such a high level of clustering may be rooted in the assumption that many authors work in the same research field, and as a consequence, are aware of each others work [13]. Another possible explanation may be that authors tend to write papers with colleagues from the same institution. Furthermore, we stress that our way of constructing a network of collaborations between authors emphasizes manuscripts with a large number of authors. Specifically, a set of authors that penned a manuscript together is represented as a clique, a graph that has a clustering coefficient of 1. Consequently, manuscripts with many authors potentially introduce a bias toward strongly clustered networks.

Another network parameter that well reflects the underlying topology of an emerging network over time is the Strong Giant Component, SGC , defined as the greatest connected subset of nodes in a network. In particular, a high value of SGC points to the observation that the vast majority of scientists are connected through mutual collaborations. During the first years of the conference (up until 2007), Table 2 indicates that the size of SGC 's was small,

Year	N	$\langle k \rangle$	C	$\langle d \rangle$	SGC	D
2000	63	1.81	0.47	1.00	9.52%	1
2001	129	2.51	0.55	1.00	6.20%	1
2002	202	2.62	0.55	3.20	10.40%	6
2003	268	2.86	0.55	3.22	8.21%	6
2004	400	2.92	0.58	4.14	10.75%	10
2005	522	3.18	0.59	3.96	14.75%	9
2006	625	3.18	0.60	4.34	14.72%	10
2007	756	3.34	0.62	4.85	20.24%	11
2008	884	3.44	0.64	7.72	41.18%	17
2009	1041	3.58	0.65	8.13	46.11%	18
2010	1170	3.70	0.66	6.60	48.55%	15
2011	1339	3.76	0.67	6.47	53.70%	15
2012	1442	3.94	0.68	5.82	58.46%	14
2013	1548	4.03	0.69	5.74	61.18%	13
2014	1683	4.14	0.70	5.52	60.90%	13

Table 2. We show properties of the cumulative authors' collaboration networks, combining manuscripts up to a given year. In particular, N is the number of nodes, $\langle k \rangle$ is the mean degree, C is the clustering coefficient. Furthermore, $\langle d \rangle$ is the avg. shortest path of the SGC , which stands for the size (percentage of nodes) of the strong giant component, while D is the diameter of the SGC .

suggesting that collaborations between authors appeared rather scattered. However, the size of the SGC doubled in 2008, indicating an increased convergence where previously present authors increasingly published a manuscript together. On the other hand, the observed increase in size also points to a gradual increase in the mean shortest path $\langle d \rangle$ between all pairs of nodes in the SGC . A closer look at our data confirmed that the increase in size of the SGC was the consequence of a merger of the two largest components from the previous year. Notably, this topological change was caused by a small set of nodes that bridged the previously disconnected components in the underlying network. As a consequence, the topological mean shortest path lengths between nodes increased substantially since shortest paths between nodes that were placed in previously disjoint components run through the small set of connecting nodes. Such an assumption is further confirmed by the increasing diameter of the underlying networks defined as the maximum of shortest paths through a given network (Table 2).

3. RESEARCH TOPICS

The analysis of the time evolution of research topics is a valuable asset for a research community to solve initial problems and to adapt to challenging areas of research. In this section, we automatically extract underlying topics from the text content of proceeding papers, allowing us to map the evolution of these topics since the inception of the MIR field.

topic	00	01	02	03	04	05	06	07	08	09	10	11	12	13	14
MIR Data & fundamentals															
mus. signal processing	17.1	-	-	-	-	-	8.0	-	-	19.5	-	-	10.9	10.2	12.3
metadata, & semantic web	11.4	5.6	-	17.0	12.5	11.5	16.1	12.7	10.5	9.8	11.8	9.8	-	-	-
social tags & user gen. data	-	-	-	-	-	-	-	13.5	10.5	12.2	10.9	12.8	11.9	12.2	-
lyrics & genres & moods	-	-	-	-	-	-	-	-	11.4	11.4	-	10.5	9.9	-	11.3
Domain Knowledge															
comp. music. & ethnomus.	-	8.3	-	-	-	-	-	-	-	-	-	-	-	-	-
mus. notation	-	8.3	-	-	-	-	-	-	-	-	-	-	-	-	-
mir & cultures	-	-	-	-	-	-	-	-	-	-	-	9.8	-	10.2	-
Mus. Features & Properties															
melody & motives	11.4	-	11.3	8.5	8.7	-	9.2	11.9	-	-	-	11.3	12.9	-	-
harmony, chords & tonality	-	13.9	-	-	13.5	8.8	9.2	10.3	9.5	13.0	10.9	10.5	11.9	10.2	-
rhythm, beat, tempo	-	19.4	-	12.8	13.5	12.4	-	-	13.3	8.9	11.8	-	-	12.2	12.3
mus. affect, emot. & mood	-	-	-	10.6	-	-	-	-	-	-	-	-	-	10.2	-
structure, segment, & form	-	-	11.3	-	-	-	-	-	10.5	12.2	10.0	12.0	8.9	10.2	13.2
Music Processing															
sound source separation	-	-	-	-	-	-	8.0	10.3	-	-	13.6	-	14.9	12.2	11.3
mus. transcrip. & annot.	5.7	8.3	-	-	-	11.5	-	-	-	-	-	-	-	12.2	-
optical mus. recognition	-	-	-	-	-	-	6.9	10.3	-	-	-	-	9.9	-	-
align., synch. & score foll.	-	-	-	10.6	-	12.4	-	-	-	-	-	-	-	-	-
mus. summarization	-	-	7.5	-	-	-	-	-	-	-	-	-	-	-	-
fingerprinting	-	-	11.3	-	-	-	-	-	-	-	-	12.8	-	-	-
automatic classification	8.6	11.1	11.3	12.8	13.5	14.2	13.8	12.7	12.4	13.0	11.8	-	-	-	14.2
indexing & querying	22.9	13.9	9.4	10.6	7.7	9.7	9.2	-	-	-	10.9	-	-	-	-
pattern match. & detection	-	11.1	-	8.5	10.6	9.7	-	-	11.4	-	-	-	-	-	5.7
similarity metrics	-	-	-	8.5	9.6	-	11.5	8.7	-	-	8.2	10.5	-	-	-
Application															
user behavior & modeling	-	-	-	-	-	-	-	-	-	-	-	-	8.9	-	-
digital libraries & archives	11.4	-	-	-	10.6	-	-	-	-	-	-	-	-	-	-
mus. retrieval systems	-	-	22.6	-	-	-	-	-	10.5	-	-	-	-	-	8.5
mus. rec. & playlist gen.	-	-	15.1	-	-	9.7	8.0	-	-	-	-	-	-	-	11.3
mus. & gaming	-	-	-	-	-	-	-	9.5	-	-	-	-	-	-	-
mus. software	11.4	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 3. Utilizing a LDA model, we determined topic evolution over time, where topics are grouped according to the topic classification in the call for papers. Values in bold correspond to the most salient topics in each ISMIR conference edition.

3.1 Topic extraction

We automatically extract the main topics by using Latent Dirichlet Allocation (LDA) [4], a generative probabilistic model in which documents are represented as random mixtures over latent topics. Each topic is characterized by a multinomial distribution over words that form those documents [4]. As a main characteristic LDA assumes that the topic distribution has a Dirichlet prior, which not only results in a smooth distribution but also simplifies the problem of topic inference [16].

In particular, we used the MALLET implementation of LDA, a java-based package for statistical natural language processing, document classification, clustering, topic modeling and machine learning applications of text [11]. MALLET’s implementation takes a text corpus and the number of topics (k) to generate as input, and produces a list of the most relevant topics for that corpus, along with the topics’ most salient terms. Furthermore, MALLET also provides a distribution of the topics among the documents that form the corpus and includes a text pre-processing step prior to generate the topic models.

Here, we build a corpus for each set of manuscripts in the ISMIR proceedings in a given year and set $k = 10$, resembling the number of oral sessions defined by the program chairs, which typically group paper presentations by

their topic affinity. For the text pre-processing step, we removed English stopwords, considered words that were longer than 2 characters and used a combination of word unigrams and bigrams. Since topics produced by an LDA model are only described by their word distribution, we manually assigned “titles” after an inspection of the most probable terms. In particular, we used the list of topics described in the conference call for papers¹ as our basis to assign and disambiguate topic titles². We also observed that this LDA implementation was systematically producing a topic containing most of the common words in any MIR publication (such as *music*, *system*, *information*, *query*, *retrieval*). Since such topics were almost never the most salient topic of a document in the corpus we removed them from our analysis.

3.2 Topic evolution

Table 3 shows the most salient topics that appeared in the ISMIR proceedings over time, as well as a visualization of their evolution, pointing to their presence in each conference edition. Each value in Table 3 represents the percentage of papers per year whose most probable topic in the

¹ <http://ismir2015.uma.es/callforpapers.html>

² due to lack of space we made the topic distribution available online: <https://goo.gl/60mG15>

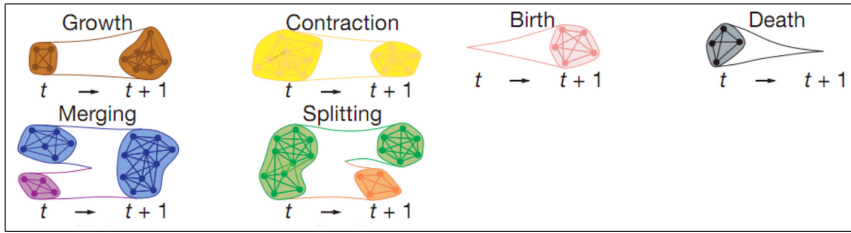


Figure 1. Fission/fusion patterns in social networks from [14]. Considering social networks over time, groups are governed by dynamic events such as mergers (i.e. fusion) and splits of groups (fission).

document–topic distribution corresponds to the topic in a given row. For instance, the topic *indexing and querying* was the most salient topic of 22.9% of the papers in the ISMIR 2000 edition. We stress that the lack of a value for a topic in a specific conference edition does not necessarily point to its absence in the underlying edition. In fact, such an observation rather indicates that the topic in question was not among the $k = 10$ most salient topics that year.

For a better interpretation, we grouped topics according to the topic classification in the call for papers. Notably, we observed that the most salient topics over time belonged to the categories “MIR Data and Fundamentals”, “Musical Features and Properties” and “Music Processing”, respectively, categories that can be regarded as the core categories in the MIR field. Some topics have been largely present over time, such as automatic classification, harmony, melody, etc. Other topics appeared or became more popular halfway through the life-span of the conference (e.g. tags, lyrics, moods, structure, etc.) or in the last few years (e.g. source separation and music cultures). Such observations may be the consequence of introducing emerging research topics or approaches from “neighboring” communities or from a shift in research funding by national or international agencies. Finally, some topics emerged that have only been present in a short time period (e.g. digital libraries or music and gaming). In particular, we highlight the *digital libraries* topic, which was more present during the first editions of the conference, but disappeared from the most salient topics over time. Such an observation may be explained by the increased focus on music content- and context-based analysis (groups 1, 3 and 4).

4. GROUP DETECTION AND EVOLUTION

In the past few years, considerable attention has been paid to uncovering topological groups in social networks. These groups are expected to fundamentally impact the network’s dynamical properties as well: nodes that belong to the same tightly connected module are expected to display highly correlated dynamical activity, compared to nodes belonging to different groups. Previous studies found that large groups are more stable and have a longer lifetime if they are capable of dynamically altering their membership, sug-

gesting that an ability to change the group composition results in better adaptability [14]. Small groups display the opposite trend, suggesting that their condition for stability is an unchanged group composition. These discoveries are expected to play a fundamental role in our understanding of human dynamics, with particular impact on our ability to detect persuasion campaigns in a changing network environment. Notably, dynamics of group composition have been noted by Dunbar and co-workers as a key mechanism to understand underlying human behavior across domains [1, 6]. In particular, we not only expect that such patterns will occur in the co-authorship networks based on conference proceedings of the ISMIR conference but also assume that the (in)stability of groups is a function of their underlying topics.

4.1 Method

Our method is a modification of the method presented in [14]. In particular, we define a co-authorship network for each edition of the conference, where each edge represents a manuscript that a pair of authors penned in a given year. Furthermore, we extract groups using the clique percolation method (CPM), an algorithm for the detection of overlapping network communities [15]. Groups in CPM, called k -clique percolation clusters, are built up from adjacent k -cliques³. Two k -cliques are considered adjacent if they share $k - 1$ nodes. Such a definition allows nodes to appear in several k -clique percolation clusters, a suitable assumption, given that authors may participate in more than one group. Specifically, we set $k = 3$, since papers in the ISMIR proceedings are co-authored on average by 3 scientists. As a consequence, this restriction implies that authors who collaborate with less than 2 other authors will never be part of a group.

After groups have been determined in a given year, we need to find their possible matches in subsequent years. In particular, we construct a joint network by merging nodes and edges of networks at consecutive time steps t and $t + 1$ [14], considering different fission/fusion patterns (Fig. 1). We label the set of groups in time t as D , the set of groups in time $t + 1$ as E , and the set of groups in the joint network

³ subgraphs of size k in which each node is connected to every other nodes

life span	1	2	3	4	5	6	8	9
num. groups	327	65	24	7	2	5	1	1

Table 4. Distribution of groups by their life time

as V . The definition of CPM implies that each group in D (E) is contained in exactly one group in V , although not all groups in V will contain a group in D (E). If a group $V_k \in V$ contains a group $E_j \in E$ but no group in D , then group E_j is considered born. Similarly, if a group $V_k \in V$ contains a group $D_i \in D$ but no group in E , then group D_i is considered dead. Furthermore, if a group $V_k \in V$ contains one or more groups in D and one or more groups in E , then the relative overlap between all different pairs (D_i^k, E_j^k) is obtained as:

$$C_{i,j}^k = \frac{D_i^k \cap E_j^k}{D_i^k \cup E_j^k} \quad (1)$$

The pair (D_i^k, E_j^k) of groups that maximizes this formula is considered a match of the same group in consecutive time steps t and $t+1$. The remaining groups are either marked as dead (D^k) or born (E^k).

Contrary to the approach in [14], we considered the overlap of nodes (instead of edges) to check whether groups in D and E are contained in V . Since our networks are built at discrete times two networks at time steps t and $t+1$ do not necessarily have a high overlap as suggested in [14]. Although the overlap of nodes might incur more noisy matchings [14], we observed that our approach is less sensitive to the noise since the overlap of networks is limited.

In some cases, we may consider a group dead in time step $t+1$ but observe it re-born in time step $t+2$ as a consequence of members that did not publish a paper in the proceedings of year $t+1$. Even though the time interval is larger than one year we consider them as the same group. Specifically, we matched such dead groups at time t with born groups at $t+n$ with $n=2$, as larger values of n only merged a very small number of groups.

4.2 Experimental results

Applying our method to our set of ISMIR proceedings we obtained a list of 432 groups, distributed as shown in table 4. Notably, we observed that only 40 groups persisted for 3 or more years, representing less than 10% of the total. In particular, we split the groups in two categories: groups with short (< 3) and a long life spans (≥ 3). Analyzing group sizes we determined the average size of each group (in terms of group members in each time step) over time and split the groups in three size categories: small (avg. size < 4 members), medium ($4 \leq$ avg. size < 5) and large (avg. size > 5). The three size categories contained 234, 120 and 78 groups, respectively.

Group size	σ_{avg}	Avg. cumul. authors
small	0.58	5.17 \pm 1.47
medium	1.18	9.06 \pm 2.54
large	2.35	13.55 \pm 3.47

Table 5. Group member variability in groups with long life time.

Group size	$\mu(\sigma)$	median
small	3.17 \pm 1.07	3
medium	3.88 \pm 1.60	3
large	6.0 \pm 2.04	5

Table 6. Topic variability in groups with long life time.

4.2.1 Group member variability

We analyzed the variability of group members when groups persisted for a longer period of time. In particular, we calculated the variance of group size for each group in each group category, and averaged them using $\sigma_{avg} = \sqrt{\sum_t var(s_t)}$, where s_t is the size of a group at a particular time t . Moreover, we computed the average number of distinct authors that participated in the group at a given time. Table 5 indicates that larger groups tend to have a higher variability of members to persist for a longer period of time. Notably, we observed the opposite when we considered small groups, confirming results in [14].

4.2.2 Topic variability

A higher topic variability means that groups change topics constantly throughout their life time. In particular, we calculated the average number of topics covered by small, medium and large groups (Table 6). Similar to the previous experiment, we only considered groups with a life time ≥ 3 . To persist longer, large groups tend to cover more topics as exemplified by a higher topic variability, as opposed to medium or small groups. Such observations suggest that the persistence of groups does not only depend on their member dynamics, but also on the variability of research topics.

4.2.3 Group characteristics and scientific impact

Focusing on the relation between group characteristics and scientific impact we considered the number of citations of each paper, as of Google Scholar, representing an indicator of scientific impact. Specifically, we group papers by their most salient topic and only select the top 10 most cited papers in each topic, providing a total of 243 papers from 28 different topics⁴. Out of this set of 243 papers, we observed that only 137 were published by groups while the remainder was penned by one or two authors. As presented in Table 7 we observed that papers written by medium sized groups tend to get significantly more citations than

⁴ some topics are present in less than 10 papers.

Group size/lifespan	avg. paper citations	# papers
small	62.54±54.74	54
medium	113.67±107.56	37
large	64.65±40.71	46
short	73.49±67.37	95
long	85.12±83.77	42

Table 7. Relation between group characteristics (size and life span) and scientific impact. We only consider the top 10 most cited papers per topic.

other group categories. As for aspects of a group's life time, papers by groups that last longer tend to get more citations than short living groups. Such an observation may be rooted in the assumption that persisting research groups with stable members may have a higher chance of getting noticed by their peers, positively affecting their research impact. Furthermore, we stress that the distribution of citations has heavy tails [18]. As a consequence the number of citations of highly cited papers varies widely, explaining the large margin of error in our analysis.

5. CONCLUSIONS

In this paper, we analyzed the evolution of the MIR field represented by the proceedings of its most prestigious conference ISMIR over the last 15 years. Notably, we found that the co-authorship network indicated a converging field of authors as indicated by the emergence of large connected and clustered network components as well as a trend toward larger research teams. While such a trend may be rooted in the way we constructed the network of co-authorships, our results also suggest that authors that have previously published conference papers separately increasingly collaborate. Therefore, present conference contributions may be viewed as 'seeds' for future collaborations between researchers that have not yet worked together. Assuming that increasing levels of collaboration govern innovation and the development of a research field, our results indicate that the ISMIR conference is a potential driver of the Music Information Retrieval field.

Furthermore, a topic analysis revealed persistent as well as 'rising' and 'falling' research topics over the years, providing a simple assessment of ISMIR's evolution. Such an analysis allowed us to investigate the longevity as well as the salience of certain topics. Our results also indicate the emergence of novel topics that potentially may dominate the focus of conference contributions in the future. Moreover, we assumed that the evolution of topics may be a function of the underlying groups of co-authors, prompting us to analyze their composition. Notably, we found that large groups persist through higher variability of team members while small groups show the opposite behavior. Furthermore, large groups show more variability of topics as opposed to medium or small groups. While not necessarily a function of group size, such results suggest that the

variability of group composition may be the driving factor of topic variability. In particular, such results support the notion that groups composed of incumbents and newcomers have a heightened chance of success [8]. As a consequence, our results suggest that large transient groups may be the drivers for innovation given that such groups provide topic variability. In turn, the arrival of new members of a group may be accompanied by the introduction of new topics. As such, our observations also suggest that group persistence is not only a question of the variability of team members but also of research topics, ultimately providing a competitive edge.

6. ACKNOWLEDGMENTS

Mohamed Sordo acknowledges Óscar Celma, Amélie Anglade, Perfecto Herrera and Simon Dixon, with whom he collaborated a few years ago on an unpublished manuscript that influenced the first part of this paper.

7. REFERENCES

- [1] Filippo Aureli, Colleen M. Schaffner, Christophe Boesch, Simon K. Bearder, Josep Call, Colin A. Chapman, Richard Connor, Anthony Di Fiore, Robin I. M. Dunbar, and S. Peter et al. Henzi. Fission-fusion dynamics. *Current Anthropology*, 49(4):627–654, 2008.
- [2] Leana Bellanca. Measuring interdisciplinary research: analysis of co-authorship for research staff at the University of York. *Bioscience Horizons*, 2(2):99–112, 2009.
- [3] Luis M. A. Bettencourt, David I. Kaiser, and Jasleen Kaur. Scientific discovery and topological transitions in collaboration networks. *Journal of Biometrics*, 3(3):210–221, 2009.
- [4] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent dirichlet allocation. *The Journal of Machine Learning Research*, 3:993–1022, 2003.
- [5] Chaomei Chen, Yue Chen, Mark Horowitz, Haiyan Hou, Zeyuan Liu, and Don Pellegrino. Towards an explanatory and computational theory of scientific discovery. *Journal of Informetrics*, 3(3):191–209, 2009.
- [6] Robin I. M. Dunbar. Social cognition on the internet: testing constraints on social network size. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 367(1599):2192–2201, 2012.
- [7] Maarten Grachten, Markus Schedl, Tim Pohle, and Gerhard Widmer. The ismir cloud: A decade of ismir conferences at your fingertips. In *10th International Society for Music Information Retrieval Conference*, pages 63–68, 2009.
- [8] Roger Guimera, Brian Uzzi, Jarret Spiro, and Luis Amaral. Team assembly mechanisms determine collaboration network structure and team performance. *Science*, 308(5722):697–702, 2005.

-
- [9] Jon M. Kleinberg. Navigation in a small world. *Nature*, 406:845, 2000.
- [10] Jin Ha Lee, M. Cameron Jones, and J. Stephen Downie. An analysis of ismir proceedings: Patterns of authorship, topic, and citation. In *10th International Society for Music Information Retrieval Conference*, pages 57–62, 2009.
- [11] Andrew K. McCallum. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>, 2002.
- [12] Mark E. J. Newman. The structure and function of complex networks. *SIAM Review*, 45(2):167–256, 2003.
- [13] Mark E. J. Newman. Who is the best connected scientist? A study of scientific coauthorship networks. *Lecture Note in Physics-New York then Berlin-*, 650:337–370, 2004.
- [14] Gergely Palla, Albert-László Barabási, and Tamás Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, 2007.
- [15] Gergely Palla, Imre Derényi, Illés Farkas, and Tamás Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.
- [16] Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.
- [17] Duncan J. Watts and Steven H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393(6684):440–442, June 1998.
- [18] Stefan Wuchty, Benjamin F. Jones, and Brian Uzzi. The increasing dominance of teams in production of knowledge. *Science*, 316(5827):1036–1039, 2007.

A TOOLKIT FOR LIVE ANNOTATION OF OPERA PERFORMANCE: EXPERIENCES CAPTURING WAGNER'S *RING* CYCLE

Kevin R. Page[†], Terhi Nurmikko-Fuller[†], Carolin Rindfleisch[‡], David M. Weigl[†]
Richard Lewis[#], Laurence Dreyfus[‡], David De Roure[†]

[†]Oxford e-Research Centre
University of Oxford
United Kingdom

[‡]Faculty of Music
University of Oxford
United Kingdom

[#]Department of Computing
Goldsmiths, University of London
United Kingdom

first.last@oerc.ox.ac.uk

first.last@music.ox.ac.uk

first.last@gold.ac.uk

ABSTRACT

Performance of a musical work potentially provides a rich source of multimedia material for future investigation, both for musicologists' study of reception and perception, and in improvement of computational methods applied to its analysis. This is particularly true of music theatre, where a traditional recording cannot sufficiently capture the ephemeral phenomena unique to each staging. In this paper we introduce a toolkit developed with, and used by, a musicologist throughout a complete multi-day production of Richard Wagner's *Der Ring des Nibelungen*. The toolkit is centred on a tablet-based score interface through which the scholar makes notes on the scenic setting of the performance as it unfolds, supplemented by a variety of digital data gathered to structure and index the annotations. We report on our experience developing a system suitable for real-time use by the musicologist, structuring the data for reuse and further investigation using semantic web technologies, and of the practical challenges and compromises of fieldwork within a working theatre. Finally we consider the utility of our tooling from both a user perspective and through an initial quantitative investigation of the data gathered.

1. INTRODUCTION AND MOTIVATION

The performance of a fully staged opera is perhaps the richest form of production when considering the potential for a wide diversity of music information united around a single body of work. Its study provides both opportunity and challenges for gathering, organising, retrieving, and analysing data and artefacts from and about the event. Thanks to a willing partnership with the Birmingham Hippodrome and the Mariinsky Opera under the baton of Valery Gergiev, their performance of all four operas compris-

ing Richard Wagner's *Der Ring des Nibelungen* (henceforth *Ring*) over five days in November 2014 presented a unique opportunity to develop and trial a musical performance annotation kit providing a structured frame of reference for interpreting collections of multimedia data.

In this paper we report on the design and implementation of the annotation software and supporting tools, which were co-designed with a musicologist to provide maximal utility when deployed for fieldwork in a working theatre. We begin by considering motivations from the fields of musicology and Music Information Retrieval (MIR).

1.1 Musicological motivation

In recent decades, methodological shifts such as a 'performative turn', widely affecting research in the Arts, Humanities and Social Sciences, and reception theory questioned musicology's traditional focus on the work as an idealised concept and on the written score. Instead, music is considered as a continuous cultural practice, couched within the respective contexts in which it is perceived, which attaches an increased value both to performance as a general concept or ritual as well as to specific performance events [6]. The individual realisation of a work in performance, especially in music theatre, differs significantly from the abstract aesthetic concept captured in the score: while the musical dimension may be treated with a high degree of 'faithfulness', scenic interpretation is created afresh in every new staging. Even in cases such as Wagner's music dramas, in which music and scenic events are coordinated down to the smallest detail, the degree to which his scenic instructions are followed varies considerably, and the reality of individual stagings goes far beyond the concept in the score. This raises the question of how a music-dramatic performance, as an ephemeral phenomenon, can be captured [14]. Analyses of recorded performances are almost as old as the respective technologies themselves [9]; but as the recording often assumes the status of an aesthetic text in the process, ephemeral phenomena are again overlooked [5]. As an audiovisual recording is neither an objective nor an exhaustive documentation, the investigation of new ways of capturing different kinds of performance data is a worthwhile undertaking. Live annotation of a performance helps to overcome the 'recording bias' by en-



© K. R. Page, T. Nurmikko-Fuller, C. Rindfleisch, D. M. Weigl, R. Lewis, L. Dreyfus, D. De Roure.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** K. R. Page, T. Nurmikko-Fuller, C. Rindfleisch, D. M. Weigl, R. Lewis, L. Dreyfus, D. De Roure. "A toolkit for live annotation of opera performance: Experiences capturing Wagner's *Ring* Cycle", 16th International Society for Music Information Retrieval Conference, 2015.

abling researchers to document events and gather information which cannot be reified in audiovisual recordings. The method of documentation and the resulting record is moreover of a significantly different quality: live-annotation allows for a selective, focused and structured record-keeping, where different annotation schemes can be tailored to a specific research question, thus integrating documentation with on-the-fly analysis. While digital technologies ease gathering of this information, this comes at a scale greater than could be recorded ‘by hand’. The ability to semantically structure gathered data for publishing and reuse, and to undertake computationally assisted analysis, provides further breadth to the study of performances.

1.2 MIR motivation

A second motivation is the utility of a well-described and structured multimedia dataset, annotated by an expert musicologist, and tooling to create such corpora, to inform, refine, and test MIR algorithms. A comprehensive data source could act as an authoritative ground truth for a variety of MIR tasks including: automated identification of musicological facets and melodic phrase recognition (e.g. leitmotif detection); tempo prediction and score following (based on page turns and annotations). It holds prospects for hypothesis-driven exploration of bio-sensed data measured from audience members, and for calibration of automated prediction of listener arousal from scores, and potentially that of musical expectation and mood.

2. RELATED WORK

The complementary nature of *performance studies* and *empirical musicology* (§6) has been noted by Cook [6]. Ker-shaw [14] discusses performances as “site-specific spectacles”, reporting research that largely confirms theatre reception as extensively influenced by idiosyncratic observer perspectives. The recording as the most accurate capturing of the live performance has been commented on by Trezise [24], while Doğan-tan Dack [8] used video recordings for a performer-centred study of chamber music, although not extending to theatrical aspects and staging.

In Section 3.3 we capture the scenic elements of performance through annotation. The extensive ethnographic study of *musical annotations* carried out by Winget [25] illustrates existing precedent for this approach, though from the perspective of musicians marking a score rather than a musicologist annotating a live operatic performance. Our technique is strongly guided by established rehearsal practice for opera, where the scenic aspects and stage directions that constitute a new staging are captured in an annotated score. These ‘scripts’ are not usually published, and no other works capturing timings of specific features of a live performance are known to the authors.

An overview of *digital technologies* in performance studies by Marsden [15] contends that research successfully bridging musicology with the digital is found within the domain of music information retrieval, rather than musical or performance analysis. For exploratory analysis, Dolan

et al. refer to Sonic Visualizer [3, 11], but are exclusive of staging, theatrics, and actor dynamics, the digital annotation of which has little prior work. Okumura *et al.* [18] modelled ways to capture deviations from strict interpretations of the score during a performance – a potential use case for our dataset. A model for acquiring content, description of data, and subsequent evaluation that complements our work is being outlined by Repetto and Serra [21].

Reflecting on corpora containing live performance and annotations, Bainbridge *et al.* [1] list The Hathitrust Digital Library [4], and the International Music Score Library Project (IMSLP) [16], as examples of large-scale digital libraries for or including music and music-related data; Doerr *et al.* comment on the role of metadata for digital library resource retrieval [10]; cross-cultural approaches and models for resource discovery in music digital libraries have been examined by Hu *et al.* [13] and Porter *et al.* [20] respectively; and Smith *et al.* [23] designed and implemented a large database for structural annotation. These inform our ontological structures (Section 3.2).

3. DESIGN AND IMPLEMENTATION

The Musical Score Annotation Kit – ‘MuSAK’¹ – was assembled from off-the-shelf hardware and applications combined with additional bespoke software, for recording the ephemera of live performance, as motivated in Section 1.1. It was designed to three primary requirements: (i) an interface sufficiently intuitive and fast enough to operate so that the musicologist could annotate under the pressure of a live performance, including turning pages to match activity on stage; (ii) for reconfigurability to incorporate changing annotation techniques and structures developed in the course of preparatory study prior to the performance events; and (iii) to be adaptable to the uncertainties of fieldwork in a working theatre environment, including potential changes to locations, power supply, access, etc. and extremely limited ‘dress rehearsals’ with a touring production.

3.1 Toolkit components

3.1.1 Annotation server and tablet interface

At the heart of MuSAK is an annotation system used by the musicologist during the performance. Initial designs called for a taxonomic palette of symbols that could be selected on an iPad tablet touchscreen and placed as annotations onto a digital copy of the score. This quickly raised three problems: (i) all proposed user interface sketches for selecting one of many annotations were complex and intrusive enough to interrupt score following and the performance observation; (ii) the operational cognitive load was judged high and different enough from traditional ‘paper and pencil’ marking to require a significant period of learning and training before use at a live event; (iii) pre-determining an adequate set of music and scenic symbols required several weeks’ precursory study, leaving limited time to add symbols to the system; furthermore, symbols might be created ad-hoc during use.

¹ <http://www.transforming-musicology.org/tools/metaMuSAK>

A pragmatic compromise was reached: short piano score pages from the IMSLP music library² were shown on a tablet, allowing freehand digital annotations according to a pictogram key of the musicologist's design. Desirable 'in content' semantics were lost, but a user experience strongly matching the traditional and familiar pattern of score marking was gained. It retained digital advantages including timestamped annotations and ease of saving, replacing, modifying, and deleting content. Image layers were 'flattened' to combine scores and existing annotations into new images, which could be redeployed for re-annotation.

A Union Platform³ server with custom room module was run on a laptop deployed in the theatre, handling storage and communication of the annotation events. To simplify distribution and quick modification, each score page was served to the tablet as a JPEG resource from an HTTP daemon, alongside client HTML and Javascript communicating with Union from the regular Safari web browser.

The web client implements buttons to turn pages and undo annotations; all annotations were recorded using Javascript event handlers to millisecond accuracy and stored both by the browser and by Union which logged to file⁴. The tablet and server were networked using a small battery powered wireless router with a private IP address space.

3.1.2 Digital pen

The tablet tool is, by necessity and design, reductionist. Inevitably some elements of the live performance are worthy of note, but either not preconceived within the symbolic key, or so unique as to require a longer form description. To accommodate this the kit includes a Livescribe Echo⁵ digital 'smart pen'. It has a standard ballpoint pen tip, but when used in conjunction with 'Anoto' paper, captures a digital copy of all writing – the paper is printed with a faint non-repeating pattern, which is read by a small infra-red camera in the pen that ascertains nib position within and between pages. While 'in content' semantics are not automatically decoded, the use of the pen is similar to standard note taking, and thus minimises intrusiveness. The digital transcription is downloaded from the pen using USB.

A second feature is a microphone for timestamped audio recordings. When polite to speak aloud, short audio comments were taken in lieu of written notes; when silence was required, the microphone captured background noise for the duration of the performance. While low quality, the latter is sufficient to calibrate temporal synchronicity.

3.1.3 Score following and replay tool

We developed a second, simple, Web application for following and recording the page-turns during the performance by a second operator, independent to the annotator (who might skip forward and backwards between pages to add

notes during quieter spells). The score-following page turns capture timings for the realization of the music contained on each page for this specific performance. Pages of the score are rendered one at a time, timestamped in a PostgreSQL database when the user advances to the next page.

An extension of this interface displays tablet annotations (§3.1.1) in real-time using an HTML canvas superimposed over the score. Data is converted from CSV to JSON to ease JavaScript working and a custom renderer calculates the appropriate time delta before drawing a stroke. JSON page-turn timestamps were also combined with the score, turning pages at the correct moment.

3.1.4 Audio and video

As is typical in commercial theatres, audio and video feeds of the performance were available within the venue (§4) but, also typically, limited distribution rights preclude their inclusion in public archives. It is desirable, and for some calculations essential, to reference their implicit existence, particularly when synchronising captured annotations for replay (§3.1.3 & §4) and structured data dissemination (§3.2) – or rather, to explicitly reference the timeline against which the notional recording was made⁶. For replay of annotations it is possible to include a *substitute* audio recording of an alternate performance (§4).

A second distinct video use was recording the annotation actions of the musicologist, providing a contextual reference for toolkit evaluation and, should the Union server fail, potential for reconstruction of annotation times.

3.2 Data publication

The use of semantic technologies to publish performance metadata from the Internet Archive Live Music Archive⁷ is described by Bechhofer *et al.* [2], and in the context of diversifying and enriching music information retrieval by Page *et al.* [19]. Crawford *et al.* [7] examines the potential of Linked Data for early music corpora, and Bainbridge *et al.* [1] comments on the effect of musical content analysis and Linked Data in the context of digital libraries. Sebastian *et al.* [22] report on ontology creation for musical performance, forms and structures.

Adopting these motivations, and to provide a strong foundation for the further investigation and reuse for musicology and MIR, we have structured our data as RDF. This entails complex ontological structures to fully and explicitly represent the items and their relationships, illustrated in Fig.1 by the timeline patterns required to encode the apparently simple relationship between the annotation of score pages and their performance on stage⁸.

A second benefit of web technology is fidelity of access at the resource level. For example, we might publish the overall structure and formal annotations, but restrict access to the video to individually registered ethnographers.

² <http://imslp.org/>

³ <http://www.unionplatform.com/>

⁴ 1 CSV file per score page with co-ordinate defined paths tracing the annotations, 1 row per straight line. Each annotation path may be described by multiple lines; paths within the same 'pen down'-'pen up' event are given the same timestamp. Pages without annotations are empty but still timestamped to record page turn times.

⁵ <http://www.livescribe.com/uk/smartpen/echo/>

⁶ While the recording is not technically required in addition to the *timeline* of the recording, its conceptual, if not actual, inclusion can simplify the metadata encoding structures and increase their comprehension.

⁷ <https://archive.org/details/etree>

⁸ See [17] for a detailed description of Linked Data generation.

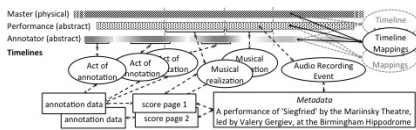


Figure 1. Simplified data modelling.

3.3 Score preparation and musicological annotations

Central to the annotation workflow, and used in several tools, the score page images required several iterations of processing⁹ and annotation before distribution in the kit. Piano and vocal arrangements were chosen to reduce the frequency of page turns, converted from IMSLP PDFs to images¹⁰. Screen use was reduced by a semi-automated process: whitespace detection identified edges, and markers indicating proportionately smaller margins were accepted or adjusted in a simple editing view; saved geometries enacted the crop and enable later scaling of annotations for overlay on original scores. Artifacts from the pre-IMSLP scanning process were cleaned and score images sharpened. Scripts applied a consistent naming scheme for images, used later for inter- and intra-opera page ordering.

A two stage annotation process reduced the note-taking required during the live performance to a minimum. Annotations extracting certain musical points of interest in the score (such as leitmotifs and marked changes of tempo or dynamics) were made by hand, using symbols designed prior to the performance. Each opera score was marked over an average of three days¹¹ and scanned to images, creating the first derivative layer.

The musicologist used the same symbolic key to annotate this layer using the tablet during the live performance, making further notes where musical aspects differed from those expected and previously marked, and of ‘stage directions’ (such as lighting, use of props, actions and movements of the characters) which were not directly marked in the original score yet are critical for any interpretation of the performance. These live annotations were ‘flattened’ into a second derivative image layer of the score.

4. TOOLKIT DEPLOYMENTS

The kit was deployed by a musicologist and two technical assistants as part of a larger project team for the Mariinsky Opera’s production of the *Ring* at the Birmingham Hippodrome. Installation in a working theatre hosting a large touring troupe¹² necessitated quick adaptations to the limitations of the available spaces and ad-hoc solutions as events unfolded – the majority beyond the control of the annotation team. Earlier design decisions to reduce

the technical complexity of the components proved worthwhile – it is not an understatement to report that a more sophisticated version would have been insufficiently resilient to the challenges of this fieldwork.

For the first night’s opera (*Das Rheingold*) the musicologist was located in a dressing room backstage with an audio and video feed from the stage; while the quality of this viewing was far from ideal, it enabled spoken Live-scriptive annotations¹³. On subsequent nights (*Die Walküre*, *Siegfried*, & *Götterdämmerung*) the ‘audio describing’ room was used, adjoining a lighting gallery rear of the circle and with an unobstructed view of the stage. In this improved location lights were dimmed and silence maintained; notes were written, not spoken. The annotation server and router were co-located with the musicologist every night and a video camera recorded the annotation process. The score-following annotation system was run from a laptop in a theatre office with an audio feed provided for the operator.

While the simplified design generally paid dividends, there were some malfunctions: we had not expected nor tested for the hour long *second* interval in *Die Walküre* and the connection between tablet and annotation server timed out. The most practicable solution was to restart both tablet and server, losing annotations for the first scene of the third act¹⁴. A second issue occurred when paging through tablet annotations after a performance, causing time-stamps to be rewritten – original times were reconstructed from page turn logs and intra-page timings.

The captured *Ring* totalled 15 hours, consisting four nights’ performance over five days, with corresponding tablet activity of over 100,000 strokes making 8,216 annotations and almost 1,300 performance based page turns. The kit deployment and data capture generated 1,316 digital images, 104 pages of writing producing nearly 13 hours of digital pen replay, and 15 hours of video footage. While Network Time Protocol (NTP) clients were used to synchronise equipment clocks some drift was observed, due to differences in Operating Systems and many devices lacking a live connection to an NTP server; these offsets are crucial for data replay and thus explicitly recorded for data publication (§3.2).

A second deployment of the kit demonstrated its flexibility in reconfiguration: at a public engagement event, audience members used their mobile devices to provide annotations while listening to a live audio replay, either by annotating musical score, or “annotating” by placing marks on a simple image with zones for e.g. fast/slow, loud/soft. Both versions of the interface were provided using simultaneous client connections. Comparative visualisations were played to a substitute audio track, derived from a commercial recording using the MATCH Vamp plugin¹⁵ and the rubberband audio time warping tool¹⁶.

⁹ The digital processing scripts described here were implemented using Image Magick and the Perl Image::Magick module.

¹⁰ An image file per page; served by respective web servers in the kit.

¹¹ For context, the score for *Das Rheingold* (the shortest) is over 250 pages; *Götterdämmerung* (the longest) comprises 365 pages.

¹² Whose predominant language was Russian, compared to English for the annotation and Hippodrome teams!

¹³ In German, the musicologist’s native tongue.

¹⁴ Which includes the section popularly known as the *Ride of the Valkyries*. The cause of this problem was not indicated in logs; rebooting may have destroyed debugging evidence.

¹⁵ <https://code.soundsoftware.ac.uk/projects/match-vamp>

¹⁶ <http://breakfastquay.com/rubberband/>

Opera	Shapes /page	S.D.	Duration	S.D.	Over-head	S.D.
Das Rheingold	5.46	4.6	34.92	15.78	-0.028	5.83
Die Walküre	6.95	6.42	44.31	26.59	0.038	8.62
Siegfried	5.76	5.19	39.23	19.06	0.044	12.97
Götterdämmerung	7.22	5.52	43.4	25.45	-0.47	14.98

Table 1. Mean annotation shapes per page, page performance durations and annotation overhead (both seconds).

5. USER EVALUATION

Post-deployment interviews with the musicologist evaluated the usability of MuSAK in this small trial according to learnability, efficiency, memorability, and satisfaction [12].

Defined as the degree of ease with which functionality can be learnt and task proficiency gained, *learnability* was evaluated through the experience of acquiring the skills necessary to complete the annotation process. The musicologist found the system non-invasive and in-line with existing annotation pragmatics, minimising training time:

“[Annotation is] very similar to the process that I as a musicologist used to do regularly...I think it worked very well because [it] fit in with actions I was very well adapted to...the tools were very non-invasive.”

Efficiency of use was measured in annotation shapes per score page and analysed through mean and standard deviation (Table 1). The average number of annotation shapes per page was between five and seven across all operas, corresponding to an average of 9.7 shapes per minute.

Memorability of the kit – the musicologist’s recall of set-up and annotation after a five month period of non-use – was assessed using a think-aloud protocol. The evaluation concluded she remembered both to a very high extent.

A qualitative evaluation assessed whether functionality and performance were *satisfactory*: the musicologist described the experience as follows, believing time needed to make additional freehand annotations and cognitively process observations made page turn annotations inaccurate.

“I was quite well able to keep up with the pace... an important realisation is that making these scenic annotations [...] requires a lot of time to think and... process even if it is only like 10 seconds or 5 seconds.”

Page turn analysis (§6) indicates that, on average, the annotator could keep pace with the performance.

The musicologist reported an ability to *capture the idiosyncratic profile of each specific performance*, including deviations from the score or expectations based on the score, as well as staging, lighting, and the behaviour of the actors. The kit was described as *supportive of traditional annotation paradigms*, not necessitating new skills for effective use, and the touchpad screen and stylus were:

“intuitive [...] similar to using pen and paper which everyone [...] analysing music is very used to.”

The additional affordances of a digital system were noted, including the automatic *capture of the temporal profile of the performance* and the *benefit of being able to easily create corrections, and undo mistakes*.

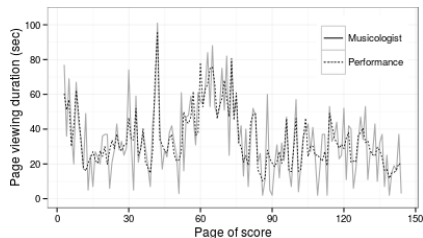


Figure 2. Musicologist’s page viewing durations and performance durations for those pages (*Siegfried* Act I).

6. DATA INVESTIGATION

A preliminary analysis considers four research questions to improve our understanding of the data captured. These come with important interpretive contexts: potentially generalisable findings are limited by the scope of data collection to a single performance of the *Ring*; annotations are recorded as continuous shapes from pen touching to leaving the screen, thus symbols comprising several distinct shapes are identified as multiple annotations; and some performance sections are excluded: the start of *Die Walküre* Act III due to kit malfunction, the first page of each opera, and those either side of the intermissions¹⁷.

6.1 Overhead of annotation task

To determine whether the overhead of annotation interfered with music following, we compared the musicologist’s page view durations with the score-page performance events. The corresponding plots reveal strong tracking of the two timelines (Fig.2); Table 1 displays the mean page performance durations, and the time difference compared to the annotator’s mean page view durations (the annotation overhead). While performance durations are variable due to changes in tempo and in musical information density on a given page, the magnitude of the mean annotation overhead is below half a second in all four performances. Standard deviations indicate there were periods when annotation acts were delayed, but overall, the musicologist was able to keep up with the music. The value is negative in two performances, indicating a tendency to read ahead.

6.2 Variability of annotation rate

We tested the variability of annotation rates¹⁸ for each night (Table 2; Figure 3). Results demonstrate significant correlations in each performance, accounting for between 18% (*Götterdämmerung*) and 43% (*Walküre*) of the variation in rank between page performance duration and number of annotation shapes per page. The finding of a largely

¹⁷ Pages were left open during the interval so durations are artefactual.

¹⁸ A hypothetical uniform rate would exhibit strong correlation between the duration of a score page performance event and the number of annotation shapes produced for that page.

Opera	Rheingold	Walküre	Siegfried	Götterdämmerung
r_s	.46	.65	.57	.42
r_s^2	.21	.43	.32	.18

Table 2. Spearman correlation ($p < .001$) between page performance duration and annotation shapes per page.

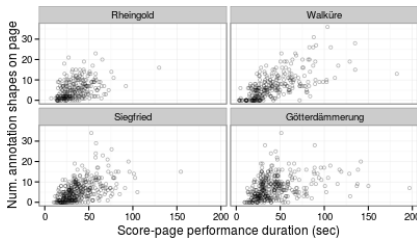


Figure 3. Correlation between score page performance duration and annotation shapes per page.

consistent annotation rate reflects the suitability of the annotation key for the task, suggesting the musicologist could adopt symbols with different granularities of meaning according to the time available. It suggests that even seemingly small ‘events’ such as gestures were not overlooked.

6.3 Annotation density; Performance correspondence

Finally, we investigated whether periods of high annotation density reflect consistent types of stage events, or corresponded to scenes of high activity or intensity. For this analysis, periods in *Siegfried* with a number of annotations per minute exceeding a threshold of the mean plus two standard deviations, as well as three major peaks in annotation activity for the final act of *Götterdämmerung*, were mapped to page sequences in the annotated score. The musicologist reinspected the corresponding pages, determining that the symbols occurring at these periods predominantly indicate changes in performers’ posture or position. These symbols largely consist of four or more shapes; thus, the high rate of annotation during these relative to other periods may be partly artefactual. These peak periods may refer to key dramatic moments, e.g. when Siegfried kisses and awakens Brünnhilde in Act III, or parts of larger dialogic scenes, where every utterance is interpolated by a postural change. Certain passages with greater staged activity (for instance when Siegfried kills Fafner) were not observed within this subset of the data. A possible explanation is that these scenes were staged with a high reliance on lighting effects, annotated with simple symbols; they were also generally drawn out for longer periods, and thus potentially overlooked by our per-minute-metric. The three peaks in the third act of *Götterdämmerung* each reflected essential moments: the Rhinemaidens telling Siegfried about the curse; Hagen killing Gunther; and Hagen struggling with the Rhinemaidens for the Ring. One other expected scene, Siegfried’s death, took up over two minutes,

and was thus represented by two rate observations that both came close to the threshold without quite meeting it. A refined measure of annotation density accounting for variations in granularity by considering the immediate temporal context could accommodate this issue.

7. CONCLUSIONS AND FUTURE WORK

We have described the software developed to, in combination with off-the-shelf hardware, form a kit used to capture data informing performance studies and the MIR analyses that may be applied to them. We have reported its use to annotate a complete production of Wagner’s *Ring* and evaluation of the kit’s performance after the deployment. An initial data driven investigation of the annotations has shown it can support and enrich analysis of the performance, and that the corpus could be developed as a ‘ground truth’ for MIR research.

Investigations to date have focused on temporal analysis of acts of annotation, whereas our next step will examine semantics within the symbols, realizing further benefits for indexing and searching within performance data. We will trial computer vision techniques to categorise pictograms in the annotation layers, and revisit options for encoding stronger symbol semantics during the annotation. While the desirable affordances of the current interface preclude full taxonomic symbol selection, our data analysis suggests even a very coarse grained categorisation (e.g. complex vs. simple events) would yield a much improved musicological understanding of the data. Our work informs future design of symbols used within the kit: ensuring greater uniformity of semantic complexity which would simplify analysis, as would the ability to more clearly delimit writing events, either by the reduction of all symbols to single (rather than compound) drawing, or through a metric combining of temporal and geometric distance. Future deployments of the kit will also record instances of ‘undo’.

Our data indicates events with complex layering of type and meaning throughout the performances, cautioning against formulation of naively phrased MIR tasks such as identifying “musicologically interesting parts in this annotated score”. Reflecting how tools can be utilised for musicology, our preliminary study makes clear there is unlikely to be a ‘perfect’ feature to automatically complete a study; instead the method is iterative, with computational analysis informed by musicology research questions and vice versa – through this iteration a fuller understanding of the question, investigation, and its limitations can be found.

8. ACKNOWLEDGEMENTS

This work was supported by the UK Arts and Humanities Research Council *Transforming Musicology* project (AH/L006820/1), part of *Digital Transformations*. The authors gratefully acknowledge assistance during the *Ring* and *Hearing Wagner* events: the Birmingham Hippodrome team, especially Zara Harris and Paul Kaynes; Mariinsky Opera Company; and the greater project team. We thank Jeff Fuller for invaluable Union Platform help and advice.

9. REFERENCES

- [1] D. Bainbridge, X. Hu, and J. S. Downie. A musical progression with Greenstone: How music content analysis and linked data is helping redefine the boundaries to a music digital library. In *Proceedings of the 1st International Workshop on Digital Libraries for Musicology in conjunction with the ACM/IEEE Digital Libraries conference 2014*, pages 1–8, 2014.
- [2] S. Bechhofer, K. R. Page, and D. De Roure. Hello Cleveland! Linked data publication of live music archives. In *14th International Workshop on Image and Audio Analysis for Multimedia Interactive Services*, pages 1–4, 2013.
- [3] C. Cannam, C. Landone, M. Sandler, and J. Bello. The Sonic Visualiser: A visualisation platform for semantic descriptors from musical signals. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 324–327, 2006.
- [4] H. Christenson. HathiTrust. *Library Resources & Technical Services*, 55(2):93–102, 2011.
- [5] N. Cook. Methods for analysing recordings. In *The Cambridge Companion to Recorded Music*, pages 221–245, 2009.
- [6] N. Cook. *Beyond the Score: Music as Performance*. Oxford University Press, Oxford, 2014.
- [7] T. Crawford, B. Fields, D. Lewis, and K. R. Page. Explorations in linked data practice for early music corpora. In *Proceedings of the IEEE/ACM Joint Conference on Digital Libraries*, pages 309–312, 2014.
- [8] M. Doğanatan Dack. The art of research in live music performance. *Music Performance Research*, 5:34–48, 2012.
- [9] H. Danuser. Urteil und Vorurteil im Interpretationsvergleich. *Zeitschrift für Musiktheorie*, 6(2):76–88, 1975.
- [10] M. Doerr, C. Bekiari, P. LeBoeuf, and Bibliothèque Nationale de France. FRBRoo, a conceptual model for performing arts. In *2008 Annual Conference of CIDOC, Athens*, pages 06–18, 2008.
- [11] D. Dolan, J. Sloboda, H. J. Jensen, B. Crüts, and E. Feygelson. The improvisatory approach to classical music performance: An empirical investigation into its characteristics and impact. *Music Performance Research*, 6, 2013.
- [12] X. Ferré, N. Juristo, H. Windl, and L. Constantine. Usability basics for software developers. *IEEE Software*, 18(1):22–29, 2001.
- [13] X. Hu and Y. Yang. Cross-cultural mood regression for music digital libraries. In *Proceedings of the IEEE/ACM Joint Conference on Digital Libraries*, pages 471–472, 2014.
- [14] B. Kershaw. Performance as research: Live events and documents. In *The Cambridge Companion to Performance Studies*, pages 23–45, 2008.
- [15] A. Marsden. ‘What was the question?’: Music analysis and the computer. In *Modern Methods for Musicology*, pages 137–147, 2009.
- [16] C. A. Mullin. International music score library project/Petrucci music library (review). *Notes*, 67(2):376–381, 2010.
- [17] T. Nurmikko-Fuller, D. M. Weigl, and K. R. Page. On organising multimedia performance corpora for musicological study using linked data. In *Proceedings of the 2nd International Workshop on Digital Libraries for Musicology*, pages 25–28, 2015.
- [18] K. Okumura, S. Sako, and T. Kitamura. Stochastic modeling of a musical performance with expressive representations from the musical score. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 531–536, 2011.
- [19] K. R. Page, B. Fields, B. J. Nagel, G. O’Neill, D. De Roure, and T. Crawford. Semantics for music analysis through linked data: How country is my country? In *e-Science (e-Science), 2010 IEEE Sixth International Conference on*, pages 41–48, 2010.
- [20] A. Porter, M. Sordo, and X. Serra. Dunya: a system to browse audio music collections exploiting cultural context. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 101–106, 2013.
- [21] R. C. Repetto and X. Serra. Creating a corpus of Jingju (Beijing opera) music and possibilities for melodic analysis. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 313–318, 2014.
- [22] V. Sebastien, D. Sebastien, and N. Conruyt. Annotating works for music education: Propositions for a musical forms and structures ontology and a musical performance ontology. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 451–456, 2013.
- [23] J. B. Smith, J. A. Burgoyne, I. Fujinaga, D. De Roure, and J. S. Downie. Design and creation of a large-scale database of structural annotations. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 555–560, 2011.
- [24] S. Tresize, N. Cook, E. Clarke, D. Leech-Wilkinson, and J. Rink. The recorded document: Interpretation and discography. In *The Cambridge Companion to Recorded Music*, pages 186–209, 2009.
- [25] M. Winget. Annotations on musical scores by performing musicians: Collaborative models, interactive methods, and music digital library tool development. *Journal of the American Society for Information Science and Technology*, 59(12):1878–1897, 2008.

SELECTIVE ACQUISITION TECHNIQUES FOR ENCULTURATION-BASED MELODIC PHRASE SEGMENTATION

Marcelo E. Rodríguez-López

Utrecht University
m.e.rodriguezlopez@uu.nl

Anja Volk

Utrecht University
a.volk@uu.nl

ABSTRACT

Automatic melody segmentation is an important yet unsolved problem in Music Information Retrieval. Research in the field of Music Cognition suggests that previous listening experience plays a considerable role in the perception of melodic segment structure. At present automatic melody segmenters that model listening experience commonly do so using unsupervised statistical learning with ‘non-selective’ information acquisition techniques, i.e. the learners gather and store information indiscriminately into memory.

In this paper we investigate techniques for ‘selective’ information acquisition, i.e. our learning model uses a goal-oriented approach to select what to store in memory. We test the usefulness of the segmentations produced using selective acquisition learning in a melody classification experiment involving melodies of different cultures. Our results show that the segments produced by our selective learner segmenters substantially improve classification accuracy when compared to segments produced by a non-selective learner segmenter, two local segmentation methods, and two naïve baselines.

1. INTRODUCTION

Motivation: In Music Information Retrieval (MIR), melody segmentation refers to the task of dividing a melody into smaller units, such as figures, phrases, or sections. Given that melody is an aspect of music shared by almost all cultures in the world, and that melodies are known to be memorable, many MIR systems base their functionality in melody processing. Automatic melody segmentation is hence an important preprocessing step for MIR tasks involving searching, browsing, visualising, and summarising music collections.

Scope: Research in automatic melody segmentation has been conducted by subdividing the segmentation problem into a number of subtasks, the most traditional one being segment boundary detection, i.e. automatically locating the time instants separating contiguous segments. In this paper

we focus on detecting the boundaries of segments resembling the musicological concept of *subphrase*. The musical factors influencing the perception of melodic segment boundaries are diverse [4, 8]. In this paper we focus on modelling factors related to previous listening experience and melodic expectation [1, 9, 23, 25, 27].

Terminology: We use the term ‘phrase’ to refer to a sequence of notes lasting roughly from 6 notes to 8 bars. We use the term ‘figure’ to refer to a relatively short sequence of notes, lasting roughly from 2-6 notes. We use the term ‘subphrase’ to refer to melodic figures in the context of phrases, i.e. as the constituent parts of a melodic phrase.

Assumptions: Our main assumption is that human listeners exposed to melodies of a given culture acquire a vocabulary of melodic figures through ‘incidental’ learning,¹ and that this acquired melodic vocabulary aids the segmentation of phrases into subphrases.² We refer to such a listener as ‘enculturated’.

Problem statement: At present, automatic melody segmenters that model previous listening experience usually do so by storing information indiscriminately into memory. We argue that selective (rather than indiscriminate) information acquisition is necessary to simulate enculturation. We hence propose and investigate two techniques for selective acquisition in the context of phrase segmentation: one in which an artificial learner selects the subphrases that give it the ‘clearest’ possible ‘understanding’ of a phrase, and another in which the learner attempts to use subphrases it ‘knows well’ to expand its melodic vocabulary. To compare the segmentations produced by enculturated segmenters using selective and non-selective acquisition techniques, we perform a melody classification experiment involving melodies of different cultures, where the segments are used as classification features.

Paper contributions: We have three main contributions. First, the proposed techniques for selective acquisition are, to the best of our knowledge, novel in the context of melody segmentation. Second, we focus on subphrase level segmentation, which is a neglected area in music segmentation research. Third, our results show that the segments produced by our selective learning segmenters substantially improve classification accuracy when compared to segments produced by using a non-selective learning



© Marcelo E. Rodríguez-López, Anja Volk.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Marcelo E. Rodríguez-López, Anja Volk. “Selective Acquisition Techniques for Enculturation-Based Melodic Phrase Segmentation”, 16th International Society for Music Information Retrieval Conference, 2015.

¹ We use the term incidental to mean that the listener does not have an explicit learning intention.

² Refer to [15, 16, 24] for experimental work in music cognition and cognitive neuroscience that supports our assumption.

segmenter, two local segmentation methods [5, 6], and two naive baselines.

Paper summary: The remainder of this paper is organised as follows: §2 reviews related work, §3 describes our selective acquisition learning model, §4 describes our proposed enculturated segmenter, §5 describes the classification experiment and presents results, §6 discusses the evaluation results, and finally, §7 summarises our conclusions and outlines possibilities of future work.

2. RELATED WORK

Previously proposed melody segmenters that model listening experience have mostly used non-selective learners. For instance, [23] presents a segmentation model with a long-term memory (LTM) component. To train the LTM model the prediction by partial match (PPM) algorithm [21] is used, which gathers and stores ngrams and ngram statistics indiscriminately into LTM. Much of the work carried out by the authors of [23] in melodic learning has focused mostly on dealing with melodic multidimensionality [19] and on the combination of short-term and long-term memory models [20], but not much attention has been paid to the construction of the LTM itself.

We base our approach on [11], where selective acquisition learning is used for motivic pattern extraction from a corpus of melodies. Our approach extends their work by proposing and testing different selective acquisition techniques, and by combining the learning approach proposed in [11] with characteristics of the ‘feature selection’ learning approach proposed in [3] for natural language processing. Moreover, we focus on using selective learning to create more powerful LTM models for melody segmentation. In the following section we describe our approach in detail.

3. ENCULTURATION VIA SELECTIVE ACQUISITION LEARNING

The goal of selective acquisition learning is to construct an enculturated LTM model. In this paper we model enculturation as a refinement process. That is, our learner takes two inputs: (1) a LTM model, which is simply a collection of melodic figures acquired during prior listening experience, and (2) a corpus of melodies of a given culture to which the learner is to be exposed. The output is a LTM model in which, ideally, only melodic figures characteristic of the culture to which the learner has been exposed are preserved. Our learning approach is summarised as pseudo code in Algorithm 1.

As shown in Algorithm 1, our learner ‘listens’ to each melody one phrase at a time, and decides which figures to store in LTM by evaluating different segmentations. That is, the learner stores in LTM only the figures that allow it to segment the phrase in an optimal way. This process is continued until the learner has acquired the melodic vocabulary that allows it to perform optimal segmentations. In the following sections we describe each part of the approach in more detail.

Input: LTM model, Phrase-segmented Melodic Corpus,

Output: LTM model

```

while termination condition not met do
  read melody from corpus;
  for each phrase in melody do
    Compute possible segmentations;
    Select the optimal segmentation;
    Store suphrases in LTM;
  Check termination condition;

```

Algorithm 1: Selective Acquisition Learning

3.1 Input/Output

3.1.1 Input: Melody Representation

Our learner takes as input melodies represented as a sequence of chromatic pitches, constrained to a range of two octaves.³ Formally, we take $p = p_1 \dots p_N$ to be a sequence of pitch intervals, where each interval $p_i \in \mathcal{A} = \{-12, \dots, 0, \dots, +12\}$. In \mathcal{A} each numerical value encodes the distance in semitones between two contiguous pitches, and the \pm symbol encodes its orientation (ascending, descending).

3.1.2 Input: phrase segmented corpus

We assume input melodies are annotated with phrase boundaries, so that our learner can process melodies on a phrase by phrase basis, finding for each an optimal segmentation. We choose to process phrases based on cognitive constraints, as exhaustively evaluating multiple segmentations for a whole melody would break known limitations of human memory.

3.1.3 Input/Output: long term memory (LTM) model

We model LTM probabilistically using a Markov modelling strategy. Essentially this boils down to constructing a data structure to hold the number of times melodic figures up to 5 intervals appear in a corpus, and then use those counts to estimate probabilities (we go into more detail in §3.3).⁴

³ In this paper our learner and segmenters take as input symbolic encodings of melodies, i.e. computer readable representations of scores transcribed by experts (see §5.1 for more details). Symbolically encoded melodies can be represented in a variety of ways, e.g. chromatic pitch, step-leap pitch intervals, inter onset intervals, and so on. In statistical learning this multi-dimensional attribute representation of melodic events can be tackled using *multiple viewpoint systems* [7, 19]. However, using multiple viewpoints comes at expense of a considerable increase in the complexity of the statistical model architecture, resulting in an increase in processing time and space requirements, as well as lower interpretability of the model. In this paper we favour using a single melodic representation to simplify the evaluation our of segmenters, which is important considering that we evaluate our segmenters indirectly, by means of a classification experiment (see §5).

⁴ The input LTM model can also be computed by sampling from known parametric distributions, e.g. in [2] the LTM model is constructed sampling from a Dirichlet distribution. However, by using corpus statistics we can assess how different (and perhaps more suitable) are the segmentations produced by one of the learners in respect to the others when exposed to the same melodies, which is a better way to try to prove or disprove our hypothesis.

3.2 Computing Possible Segmentations

Ideally, our learner should evaluate all possible segmentations of a phrase. However, processing time is exponential on the number of notes in the phrase, so in practice evaluating all segmentations is unfeasible. Thus, we use the algorithm proposed in [17] to efficiently compute a constrained space of possible segmentations. The algorithm takes as input the minimum and maximum length of subphrases, as well as the minimum and maximum number of subphrases. As we mentioned previously we have limited subphrases to be sequences of 1-5 intervals in length. We also limit phrases to be composed of at most 6 subphrases (by doing so we are able to cope with phrases of a maximum length of 30 intervals).

3.3 Select the optimal segmentation

Below we present two techniques to select an optimal segmentation. One in which the learner selects subphrases that give it the ‘clearest’ possible understanding of a phrase, and another in which the learner uses subphrases it ‘knows well’ to increase its vocabulary.

3.3.1 Common and Complete Figures

Melodic figures that aid segmentation should be ‘characteristic’ of a melodic culture. One way to measure how characteristic figures are is by searching for ‘common’ figures in a corpus representative of a melodic culture. However, common figures are mainly of short duration, and normally less specific and informative than figures of larger duration (see [28]). There is hence a trade-off between how common a figure is and how specific to a given tradition it can be.⁵ Thus, we need a way to automatically determine how long do the figures we are after need to be, so that we search for the longest possible common figures instead of only the most common ones. One way to do so is by attempting to determine if a given figure is somehow ‘complete’ on its own, or if its part of a larger figure. Our search then would be for figures that are common, yet large enough so as to be perceptually complete. According to melodic expectation theory [14, 27], the perceptual completeness of a melodic figure is inversely proportional to the degree by which it stimulates expectation. In other words, melodic figures for which is hard to predict what comes next are perceived as more complete than those for which is easy to predict what comes next.

Using information theory we can attempt to jointly quantify the commonness and completeness of a figure. If from within a phrase of length T we take a figure $w = p_i \dots p_j$, with $i, j \in [1 : T]$, we can compute its conditional entropy h as

$$h(x|w) = P(w) \sum_{x \in \mathcal{A}} P(x|w) \log(P(x|w)) \quad (1)$$

⁵ In natural language this is also a commonly found problem, ‘content’ or informative words (e.g. nouns) tend to be of greater length than ‘non-content’ words (e.g. determinants).

where x is used to symbolise melodic events that can follow w , and P denotes probability. In Eq. 1 the first term $P(\cdot)$ will be high for common figures in a corpus, and the second term $\sum P(\cdot) \log(P(\cdot))$ will be high if it is hard to predict what comes after w . Hence, h will be high for figures that are common and complete in an information theoretic sense.

The values of probabilities $P(\cdot)$ can be estimated from the counts of w and the concatenation wx in a given melodic corpus: $P(w) \sim N(w)/N_T$ and $P(x|w) \sim N(wx)/N(w)$, where $N(\cdot)$ denotes counts, and N_T denotes the total number of counts for figures of length equal to w in the corpus.

3.3.2 Monitoring LTM

Using conditional entropy we can monitor the state of our LTM before and after a new melodic figure is listened to. So, first, the total entropy for figures w of the same size is

$$H^o = - \sum_{w \in \mathcal{A}^*} P(w) \sum_{x \in \mathcal{A}} P(x|w) \log P(x|w) \quad (2)$$

where we use \mathcal{A}^* to denote the space of all figures of size o with attribute space \mathcal{A} . In our LTM $o = \{1, \dots, 5\}$ and hence its total entropy is

$$H = H^1 + \dots + H^5 \quad (3)$$

and then we can define ΔH as

$$\Delta H = H_{\text{after listening to } w} - H_{\text{before listening to } w} \quad (4)$$

which allows us to monitor the evolution of our LTM.

3.3.3 Selection Technique 1

We have now the necessary information to formulate our first selection technique. Since common and complete figures are expected to have high entropy, a ‘good’ phrase segmentation among a group of possible segmentations is that segmentation with the highest average ΔH . That is, if we have a space of possible segmentations \mathcal{S} , the average ΔH of a candidate segmentation $s = w_1, \dots, w_m$ is

$$\phi(s) = \frac{\Delta H(w_1) + \dots + \Delta H(w_m)}{m} \quad (5)$$

and hence our first selection technique is

$$s^* = \operatorname{argmax}_{s \in \mathcal{S}} \phi(s) \quad (6)$$

Where s^* denotes the segmentation with maximal score. Note that, to ensure convergence, the learner stores in LTM only the subphrases in s^* for which ΔH is positive.

One problem with our first technique is that it makes our learner very conservative. The melodic figures stored are characteristic of the corpus as a whole. Hence, the technique operates under the assumption that the corpus is stylistically homogeneous. For most cultural traditions the assumption of complete stylistic homogeneity is too strong (it is likely that certain figures are important but only characteristic of subsets of the corpus).

Collection Name	Subset Name	Cultural Origin of Sample	Encoding	Number of Melodies	Average Melody Size in Notes	Number of Phrases	Average Phrase Size in Notes
MTC	FS	Dutch	**kern	4120	52.3 (22.5)	19935	9.1 (2.5)
EFSC	CHINA	Chinese	**kern	2201	62.8 (41.2)	11046	12.5 (4.7)
OHFT	-	Hungarian	EsAC	2323	38.6 (12.0)	9308	9.6 (3.2)

Table 1. Melodic Corpora. Numbers in parenthesis correspond to standard deviation.

3.3.4 Selection Technique 2

Our second technique aims to relax the assumption of homogeneity and stimulate the learner to expand its vocabulary. More importantly, it aims to reveal segmentations in which one or more subphrases are common and complete, and others are representative of the melody, yet relatively rare in the corpus. For a figure w the latter idea can be quantified as

$$\rho(w) = -P_{melody}(w) * \log(P_{corpus}(w)) \quad (7)$$

with $P_{melody}(w) \sim M(w)/M_T$ and $P_{corpus}(w) \sim N(w)/N_T$, where M denotes counts of w in the melody, M_T is used to indicate the total number of counts of figures of size equal to w in the melody/corpus, and N denotes counts of w in the corpus.

For a complete segmentation we take the average of ρ

$$\bar{\rho}(s) = \frac{\rho(w_1) + \dots + \rho(w_m)}{m} \quad (8)$$

Finally, we combine the $\bar{\rho}$ and ϕ using a geometric mean:⁶

$$\lambda(s) = \sqrt{\phi(s) \cdot \bar{\rho}(s)} \quad (9)$$

and compute our second technique as

$$s^* = \operatorname{argmax}_{s \in \mathcal{S}} \lambda(s) \quad (10)$$

Where s^* denotes the segmentation with maximal score. Our learner stores all subphrases of s^* in LTM.

3.4 Termination Condition

We keep track of the scores of s^* when processing the corpus, expecting that, as the learner reaches convergence, the score difference between subsequent instances s^* gets smaller and smaller. We hence assume convergence has been reached if $\Delta s^* < \epsilon$.

Since Eq. 10 encourages learning new vocabulary, convergence is slow and not guaranteed. Thus, in addition to $\Delta s^* < \epsilon$, we also set a maximum number of learning iterations as a second termination condition.

4. ENLARGED SEGMENTATION

Once the LTM model has been trained (via either selective or non-selective learning), our segmenter proceeds in a way similar to Algorithm 1. That is, it processes each melody a phrase at a time, for each phrase it computes a

⁶ Since $\phi(s)$ can in principle be negative, to compute λ we consider negative $\Delta H(w)$ values to be zero when computing $\phi(s)$ to avoid the possibility of negativity.

space of possible segmentations, and selects the best one. However, this time the selection of the best segmentation is made by computing

$$\bar{h}^* = \operatorname{argmax}_{s \in \mathcal{S}} \bar{h}(s) \quad (11)$$

where $\bar{h}(s) = \frac{h(w_1) + \dots + h(w_m)}{m}$ and $h(w)$ is computed using Eq. 1.

5. EVALUATING SUBPHRASE SEGMENTATIONS

At present, freely available corpora annotated with subphrase boundaries do not exist. This implies we are unable to evaluate our segmenters in a traditional scenario (i.e. by comparing automatic segmentations to human-annotated segmentations). Hence, we opt for a ‘use-case’ evaluation scenario: test the output of our segmenters in a melody classification experiment.

The classification task consists in predicting the cultural origin of each melody in a dataset of melodies, using subphrases as classification features. In this scenario ‘good’ segmentations should facilitate classification and thus result in high classification performance.

In the following subsections we describe the melodic corpora used for our classification experiment, the compared segmenters, the classifiers employed, and finally we list evaluation metrics and present results.

All segmenters and baselines were coded in Matlab. All source files as well as the train/test data listings are available at <http://www.projects.science.uu.nl/music/>.

5.1 Phrase Annotated Melodic Corpora

The melodic corpora used in our experiments is summarised in Table 1. The *Meertens Tune Collection*⁷ (MTC) is a collection of Dutch folk songs. The *Essen Folk Song Collection*⁸ (EFSC) is a collection of vocal folk songs from Eurasia. The *Old Hungarian Folksong Types* collection⁹ (OHFT) is a collection of vocal folk songs from Hungary.

All corpora summarised in Table 1 have been annotated with phrase boundaries by expert Ethnomusicologists.¹⁰ We cleaned the collections by removing all melodies with overly short and overly long phrases. We considered a

⁷ <http://www.liederenbank.nl>

⁸ <http://www.esac-data.org>

⁹ We obtained the OHFT data directly from the author of [11].

¹⁰ In the case of the EFSC-CHINA the origin of the phrase markings is uncertain. However, it is often assumed it corresponds to notated breath marks and/or to the phrase boundaries of lyrics. In the case of the MTC-FS phrase boundary markings where produced by two experts (which agreed on a single segmentation). The annotation process is detailed in [26]. In the case of the OHFT the phrase boundary marking process is detailed in [10, 12].

Segmenter	Parameter Setting	Segmentation Results (for the best parametric setting)								
		Mean Number of Subphrases per Phrase			Mean Number of Subphrases per Melody			Total Number of Unique Subphrases per Corpus		
		C	H	D	C	H	D	C	H	D
NS	LTM training: PPM-C, with exclusion, 1000 melodies of each culture.	5.0	4.8	4.8	27.7	16.7	23.4	1300	1091	1197
ST1	LTM training: convergence 10E-8 or 8000 phrases, 1000 melodies of each culture.	3.8	3.7	3.7	21.7	13.7	19.6	3437	2562	2204
ST2	LTM training: convergence 10E-8 or 8000 phrases, 1000 melodies of each culture.	3.6	3.5	3.5	23.2	15.4	21.3	3566	2743	2311
LBDM	detection threshold {0.2, 0.4 , 0.6}, others: suggested setting in [5].	3.0	2.9	2.9	15.5	10.4	15.4	4497	3841	2999
PAT	detection threshold {0.2, 0.4 , 0.6}, others: suggested setting in [6].	3.6	3.4	3.4	19.3	11.4	16.7	3603	3139	2810
FIXLEN	constant size $CS = 3$ intervals.	4.0	3.8	3.8	22.0	13.5	18.9	1371	1179	1474
RAND	constant size $RS \in [2 - 4]$ intervals.	4.1	3.9	3.9	22.2	13.5	19.2	2827	2413	2551

Table 2. Parameter settings and segmentation results. C - Chinese, H - Hungarian, D - Dutch. Text in bold indicates best performing parametric settings.

phrase to be overly short if it contains only one note or one interval. We considered a phrase to be overly long if it is longer than 30 notes in length.

5.2 Enculturated Segmenters

We evaluate three enculturated segmenters: NS, ST1, ST2. The NS segmenter uses a LTM model trained with non-selective acquisition (using the PPM-C algorithm [22]). The ST1 segmenter uses a LTM model trained with the selective acquisition technique 1, Eq. 6. The ST2 segmenter uses a LTM model trained with the selective acquisition technique 2, Eq. 10. A sample of 1000 melodies from each collection is used to train the LTM models. The parametric settings for each enculturated segmenter are specified in Table 2.

5.3 Reference Segmenters and Baselines

We compared the performance of the enculturated segmenters to two local boundary detection segmenters (LBDM and PAT), and two naïve baseline segmenters (FIXLEN and RAND). The LBDM and PAT segmenters were selected for comparison because they have been used for subphrase level segmentation in the past [6, 18]. The LBDM segmenter [5] computes subphrase boundaries by detecting large pitch intervals and inter-onset-intervals. Intervals sizes are given a score by comparing them to immediately surrounding intervals (the larger the difference the higher the score). High scoring intervals are taken as subphrase ends. The PAT segmenter [6] computes subphrase boundaries by detecting and scoring repetitions of pitch interval sequences within each phrase. The starting points of high scoring repetitions are taken as subphrase starts. The FIXLEN baseline segments a phrase into subphrases of constant size. The RAND baseline segments a phrase into subphrases of randomly chosen sizes. The parametric settings for each of the reference and baseline segmenters are specified in Table 2.

5.4 Features and Classifiers

As mentioned above, in our experiment we are interested in evaluating the effectiveness of subphrases as classification features. To use subphrases in the most transparent

way, we represent melodies as a ‘bag-of-subphrases’. That is, we use a vector space model representation,¹¹ where each vector element is weighted using the common term frequency - inverse document frequency ($tf * idf$) heuristic [13]. We then use two simple and well known classifiers for the cultural origin prediction task: k -means and k nearest neighbours (kNN).

Segmenter	k-means (k=3)			kNN (k optimised)		
	\bar{R}	\bar{P}	\bar{A}	\bar{R}	\bar{P}	\bar{A}
NS	0.94	0.93	0.71	0.93	0.87	0.83
ST1	0.90	0.95	0.74	0.93	0.94	0.87*
ST2	0.92	0.93	0.71	0.92	0.96	0.88
LBDM	0.47	0.50	0.47	0.75	0.84	0.76
PAT	0.74	0.76	0.58	0.83	0.87	0.79
FIXLEN	0.88	0.89	0.67	0.86	0.90	0.83
RAND	0.84	0.84	0.63	0.88	0.85	0.78

Table 3. Classification results: recall (\bar{R}), precision (\bar{P}), and accuracy (\bar{A}) averaged over 10-folds. Text in bold highlights the highest performances. Asterisks indicate performances that are not significantly different from the highest performances.

5.5 Test set, Performance Measures, and Results

We constructed a dataset of 3000 melodies by randomly sampling 1000 melodies from each corpus. (All melodies used to train the enculturated segmenters were excluded from the sample.) For each of the 3000 melodies, the classifiers are required to predict whether the melody is of Hungarian, Chinese, or Dutch origin. **Validation technique:** We used 10-fold cross validation to iteratively separate the melodic dataset into training and test sets. **Evaluation measures:** Given a N_{total} of melodies per fold to be classified, we use tp to indicate the number of true positives, fp the false positives, and fn the false negatives. With these statistics we measure measure classification performance using accuracy $A = \frac{N_{correct}}{N_{total}}$, precision $P = \frac{tp}{tp+fp}$ and recall $R = \frac{tp}{tp+fn}$. **Statistical testing:** We

¹¹ in a vector space model, melodies are represented as a vector of size $|V|$, where $|V|$ is the number of unique figures occurring in the corpus. If a figure occurs in the melody, its value in the vector is equal to the number of times it appears in the melody. The frequency of occurrence of each figure is then used as a feature for classification.

used an ANOVA test ($\alpha = 0.01$) with Bonferroni correction to test the statistical significance of the differences in accuracy for each segmenter. **Setting and optimising classifier parameters:** The training sets were used to optimise the permutation labels of the k-means classifier and select the optimal number of nearest neighbours for the kNN classifier. The optimal number of nearest neighbours (selected from $k \in [1, 15]$) was set by optimizing cross-validated accuracy on the training data.

The results of our experiment are presented in Table 3. We discuss our results below.

6. DISCUSSION

6.1 Selective vs. Non-Selective Learning Segmenters

Table 2 shows the NS segmenter produces relatively short segments, resulting in an average of ~ 4.9 subphrases per phrase, and an average of ~ 1196 unique subphrases over all three corpora. Conversely, the ST1-2 segmenters produce larger segments, resulting in an average of ~ 3.6 subphrases per phrase, and an average of ~ 2767 unique subphrases over all three corpora. Using the k-means classifier with subphrases computed using ST1 we obtain a (statistically significant) 3% \bar{A} improvement over the NS segmenter, which seems to be driven by a 2% improvement in \bar{P} . Using the k-NN classifier with subphrases computed using both ST1 and ST2 we obtain (statistically significant) 3-4% \bar{A} improvements over the NS segmenter, which are again in pair with 7-9% increases in \bar{P} . These results show the larger segments produced by the ST1-2 segmenters allow better discrimination between melodies of different cultural origin, suggesting that selective learning leads to better models of prior listening experience than non-selective learning.

6.2 Selective Learning Segmenters vs. Local Segmenters

Segmentation results in Table 2 show that local segmenters prefer larger segments than the ST1-2 segmenters. Also, the local segmenters produce an average of ~ 3481 unique subphrases over all three corpora, which is 741 subphrases larger than the average of unique subphrases produced by the ST1-2 segmenters. Table 3 shows that \bar{A} results using the segments produced by ST1-2 are $>8\%$ better than \bar{A} results using the segments produced by LBDM and PAT. The \bar{A} performance improvements are in line with relatively large improvements in both \bar{P} and \bar{R} . These results show that the larger segments produced by the local segmenters leads to an increase in unique subphrases, and that these unique subphrases are not discriminative of cultural origin. The relatively large improvements in \bar{A} of the ST1-2 segmenters over the local segmenters supports the hypothesis that enculturated listening might be of importance for the segmentation of melodic phrases.

6.3 Selective Learning Segmenters vs. Baselines

Table 2 shows the baseline segmenters produce relatively short segments (of 2 or 3 intervals), resulting in an av-

erage of ~ 3.9 subphrases per phrase, and an average of ~ 1969 unique subphrases over all three corpora. When using the k-means classifier we can observe significant and relatively large differences ($> 5\%$) between the \bar{A} obtained using ST1-2 and those obtained using the baseline segmenters. These results show the larger segments produced by the ST1-2 segmenters allow better discrimination between melodies of different cultural origin than the shorter segments produced by the baseline segmenters, indicating once more the ST1-2 segmenters might be capturing important aspects of subphrase structure.

6.4 Scepticism

Any conclusions from our use case evaluation results are limited to classification schemes using ‘bag-of-subphrases’ representations of melodies. This representation limits the similarity assessment between any two subphrases to exact matches, which might be introducing an unwanted bias on the evaluation. To draw more definitive conclusions our experiment needs to be complemented with other use case studies.

7. CONCLUSIONS

In this paper we introduce techniques for selective acquisition learning in the context of melodic segmentation, specifically the segmentation of melodic phrases into subphrases. Our aim is to show that enculturated listening is important for the segmentation of melodic phrases, and that selective rather than indiscriminate acquisition techniques are better to model an enculturated segmenter. We present two selective acquisition techniques: one in which an artificial learner selects the subphrases that give it the ‘clearest’ possible understanding of a phrase, and another in which the learner attempts to use subphrases it ‘knows well’ to expand its melodic vocabulary.

To test the segmentations produced by enculturated segmenters using selective and non-selective acquisition techniques, we perform a melody classification experiment involving melodies of different cultures. Our results show that the segments produced by our selective learning segmenters substantially improve classification accuracy when compared to segments produced by using a non-selective learning segmenter, two local segmentation methods, and two naive baselines.

In future work we plan to conduct experiments to test the sensitivity of our selection techniques to cross-learning. That is, cases in which the learners have prior knowledge of one melodic tradition and are required to adapt their knowledge to the particularities of a different melodic tradition. We also plan to extend the current approach so that it can process multiple attribute representations of a melody. To this end an integration between our approach and the multiplevuepoint formalism of [7, 19] is planned.

Acknowledgments: We thank Z. Juhász for sharing with us the OHET dataset, and also to the anonymous reviewers for the useful comments. This work is supported by the Netherlands Organization for Scientific Research, NWO-VIDI grant 276-35-001 to A. Volk.

8. REFERENCES

- [1] S. Abdallah, H. Ekeus, P. Foster, A. Robertson, and M. Plumbley. Cognitive music modelling: An information dynamics approach. In *3rd International Workshop on Cognitive Information Processing (CIP)*, pages 1–8. IEEE, 2012.
- [2] S. Abdallah and M. Plumbley. Information dynamics: patterns of expectation and surprise in the perception of music. *Connection Science*, 21(2-3):89–117, 2009.
- [3] A. Berger, V. Della Pietra, and S. Della Pietra. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71, 1996.
- [4] M. Bruderer, M. McKinney, and A. Kohlrausch. The perception of structural boundaries in melody lines of western popular music. *Musicae Scientiae*, 13(2):273–313, 2009.
- [5] E. Cambouropoulos. The local boundary detection model (LBDM) and its application in the study of expressive timing. In *Proceedings of the International Computer Music Conference (ICMC01)*, pages 232–235, 2001.
- [6] E. Cambouropoulos. Musical parallelism and melodic segmentation. *Music Perception*, 23(3):249–268, 2006.
- [7] D. Conklin and I. H. Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.
- [8] I. Deliège. Wagner alte weise: Une approche perceptiv. *Musicae Scientiae*, 2(1 suppl):63–89, 1998.
- [9] D. Huron. *Sweet anticipation: Music and the psychology of expectation*. MIT press, 2006.
- [10] P. Járdányi. Experiences and results in systematizing hungarian folk-songs. *Studia Musicologica*, pages 287–291, 1965.
- [11] Z. Juhász. Segmentation of hungarian folk songs using an entropy-based learning system. *Journal of New Music Research*, 33(1):5–15, 2004.
- [12] Z. Kodály and L. Vargyas. *Folk music of Hungary*. Da Capo Press, 1982.
- [13] C. D. Manning, P. Raghavan, and H. Schütze. Scoring, term weighting and the vector space model. *Introduction to Information Retrieval*, 100, 2008.
- [14] L. B. Meyer. Meaning in music and information theory. *Journal of Aesthetics and Art Criticism*, pages 412–424, 1957.
- [15] S. J. Morrison, S. M. Demorese, E. H. Aylward, S. C. Cramer, and K. R. Maravilla. fMRI investigation of cross-cultural music comprehension. *Neuroimage*, 20(1):378–384, 2003.
- [16] Y. Nan, T. R. Knösche, and A. D. Friederici. The perception of musical phrase structure: a cross-cultural erp study. *Brain research*, 1094(1):179–191, 2006.
- [17] J.D. Opdyke. A unified approach to algorithms generating unrestricted and restricted integer compositions and integer partitions. *Journal of Mathematical Modelling and Algorithms*, 9(1):53–97, 2010.
- [18] N. Orio and G. Neve. Experiments on segmentation techniques for music documents indexing. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, pages 104–107, 2005.
- [19] M. Pearce. *The construction and evaluation of statistical models of melodic structure in music perception and composition*. PhD thesis, Department of Computing, City University, 2005.
- [20] M. Pearce, D. Conklin, and G. Wiggins. Methods for combining statistical models of music. In *Computer Music Modeling and Retrieval*, pages 295–312. Springer, 2005.
- [21] M. Pearce and G. Wiggins. An empirical comparison of the performance of ppm variants on a prediction task with monophonic music. In *Artificial Intelligence and Creativity in Arts and Science Symposium*, 2003.
- [22] M. Pearce and G. Wiggins. Improved methods for statistical modelling of monophonic music. *Journal of New Music Research*, 33(4):367–385, 2004.
- [23] M. Pearce and G. Wiggins. The information dynamics of melodic boundary detection. In *Proceedings of the Ninth International Conference on Music Perception and Cognition*, pages 860–865, 2006.
- [24] M. Rohrmeier, P. Rebuschat, and I. Cross. Incidental and on-line learning of melodic structure. *Consciousness and cognition*, 20(2):214–222, 2011.
- [25] C. Thornton. Generation of folk song melodies using bayes transforms. *Journal of New Music Research*, 40(4):293–312, 2011.
- [26] P. van Kranenburg, M. de Bruin, L. Grijp, and F. Wiering. The meertens tune collections, 2014.
- [27] G. Wiggins and J. Forth. IDyOT: A computational theory of creativity as everyday reasoning from learned information. In *Computational Creativity Research: Towards Creative Machines*, pages 127–148. Springer, 2015.
- [28] J. Wołkiewicz, Z. Kulka, and V. Kešelj. N-gram-based approach to composer recognition. *Archives of Acoustics*, 33(1):43–55, 2008.

Oral Session 1

Corpus Analysis & Annotation

CORPUS ANALYSIS TOOLS FOR COMPUTATIONAL HOOK DISCOVERY

Jan Van Balen¹ John Ashley Burgoyne² Dimitrios Bountouridis¹
Daniel Müllensiefen³ Remco C. Veltkamp¹

¹ Department of Information and Computing Sciences, Utrecht University

² Music Cognition Group, University of Amsterdam

³ Department of Psychology, Goldsmiths, University of London

J.M.H.VanBalen@uu.nl

ABSTRACT

Compared to studies with symbolic music data, advances in music description from audio have overwhelmingly focused on ground truth reconstruction and maximizing prediction accuracy, with only a small fraction of studies using audio description to gain insight into musical data. We present a strategy for the corpus analysis of audio data that is optimized for interpretable results. The approach brings two previously unexplored concepts to the audio domain: audio bigram distributions, and the use of corpus-relative or “second-order” descriptors. To test the real-world applicability of our method, we present an experiment in which we model song recognition data collected in a widely-played music game. By using the proposed corpus analysis pipeline we are able to present a cognitively adequate analysis that allows a model interpretation in terms of the listening history and experience of our participants. We find that our corpus-based audio features are able to explain a comparable amount of variance to symbolic features for this task when used alone and that they can supplement symbolic features profitably when the two types of features are used in tandem. Finally, we highlight new insights into what makes music recognizable.

1. INTRODUCTION

This study addresses the scarcity of corpus analysis tools for audio data. By *corpus analysis*, we refer to any analysis of a collection of musical works in which the primary goal is to gain insight into the music itself. Such analyses make up only a small fraction of the music computing field, with much more research being done on classification, recommendation and retrieval [16], where the focus is often more on prediction accuracy than interpretability. Examples of corpus analysis studies include work on summarization and visualisation (e.g., [1]), hypothesis testing, (e.g., evidence for Western influence in the use of African

tone scales in [11]), and discovery-based analysis (e.g., of the structural melodic features that predict performance in a music memory task [12]).

Strikingly, while audio data is by far the most widely researched form of information in the community [16], a brief review suggests that only a minority of corpus analysis studies used audio data. This includes the above work on visualisation [1], tone scales analysis [11], and a number of recent studies on the structure and evolution of popular music [10, 15, 18]. Symbolic corpus analysis, in contrast, includes Huron’s many studies [9], Conklin’s work on multiple viewpoints and Pearce’s extensions [6, 14], corpus studies of harmony [5, 7] as well as toolkits such as Humdrum,¹ Idyom,² and FANTASTIC.³

Although the music information retrieval community has made substantial progress in improving the transcription of audio to symbolic data, considerable hurdles remain [16]. We therefore aim to further the resources for audio analysis. We present a set of audio corpus description features that are founded on the use of three novel concepts. A new kind of melodic and harmonic interval profiles are used to describe melody and harmony, extending the notion of interval bigrams to the audio domain. We then propose three so-called *second-order* features, a concept that has yet to be applied to audio features. Finally, we define song-based and corpus-based second-order features.

We test our newly developed analysis pipeline in a case study on “hook discovery”.

2. CORPUS-BASED AUDIO FEATURES

2.1 Harmony and Melody Description

We propose a novel set of harmony and melody descriptors. The purpose for these descriptors is to translate basic harmonic and melodic structures to a robust representation on which corpus statistics can be computed. They should be relatively invariant to other factors such as tempo and timbre, and have a fixed size.

In [17], the correlation matrix of the chroma features is used as a harmonic descriptor. The 144-dimensional



© Jan Van Balen, John Ashley Burgoyne, Dimitrios Bountouridis, Daniel Müllensiefen, Remco C. Veltkamp.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Jan Van Balen, John Ashley Burgoyne, Dimitrios Bountouridis, Daniel Müllensiefen, Remco C. Veltkamp. “Corpus Analysis Tools for Computational Hook Discovery”, 16th International Society for Music Information Retrieval Conference, 2015.

¹ www.musiccog.ohio-state.edu/Humdrum/

² code.soundsoftware.ac.uk/projects/idyom-project

³ www.doc.gold.ac.uk/isms/m4s/

‘chroma correlation features’ measure co-occurrence of harmonic pitch. They capture more detail than a simple chroma pitch histogram, while preserving tempo and translation-invariance. The feature was shown to perform reasonably well in a small-scale cover song experiment. In this study we extend this and two related concepts to three new interval representations. Whereas pitch bigram profiles are expected to strongly correlate with the key of an audio fragment, interval bigrams are key-invariant, which allows them to be compared across songs.

The **Harmonic Interval Co-occurrence (HIC)** is based on the *triad profile*, which is defined as the three-dimensional co-occurrence matrix of three identical copies of the chroma time series $c_{t,i}$ (t is time, i is pitch class):

$$\text{triads}(c)_{i_1, i_2, i_3} = \sum_{\tau} c_{t, i_1} c_{t, i_2} c_{t, i_3}. \quad (1)$$

The pitch class triplets in this feature can be converted to interval pairs using the function:

$$\text{intervals}(X)_{j_1, j_2} = \sum_{i=0}^{12} X_{(i-j_1) \bmod 12, i, (i+j_2) \bmod 12}. \quad (2)$$

This essentially maps each triad (i_1, i_2, i_3) to a stack of intervals $(i_2 - i_1, i_3 - i_2)$. A major chord $(0, 4, 7)$ would be converted to $(4, 3)$, or a major third with a minor third on top. Applied to the triads matrix, the intervals function yields the *harmonic interval co-occurrence* matrix,

$$\text{HIC}(c)_{j_1, j_2} = \text{intervals}(\text{triads}(c_{t,i})) \quad (3)$$

It measures the distribution of triads in an audio segment, represented by their interval representation. For example, a piece of music with only minor chords will have a strong activation of $\text{HIC}_{3,4}$, while a piece with a lot of tritones will have activations in $\text{HIC}_{0,6}$ and $\text{HIC}_{6,0}$.

The same processing can be applied to the melodic pitch to obtain the **Melodic Interval Bigrams (MIB)**. We first define the three-dimensional *trigram profile* as an extension of the two-dimensional bihistogram in [17]:

$$\text{trigrams}(m)_{i_1, i_2, i_3} = \sum_{\tau} \max_{t} (m_{t-\tau, i_1}) m_{t, i_2} \max_{t} (m_{t+\tau, i_3}), \quad (4)$$

with $\tau = 1 \dots \Delta t$ and m the melody matrix, a binary chroma-like matrix containing the melodic pitch activations. The result is a three-dimensional matrix indicating how often triplets of melodic pitches (i_1, i_2, i_3) occur less than Δt seconds apart. The pitch trigram profile can be converted to an interval bigram profile by applying the intervals function (Eqn 2). This yields the *melodic interval bigrams* feature, a two-dimensional matrix that measures which pairs of pitch intervals follow each other in the melody:

$$\text{MIB}(X)_{j_1, j_2} = \text{intervals}(\text{trigrams}(m_{t,i})). \quad (5)$$

Finally, the *harmonisation feature* in [17] measures which harmonic pitches in the chroma c co-occur with the melodic pitches in the melody m . We derive a **Harmonisation Interval (HI)** feature as:

$$\text{HI}(m, h)_j = \sum_{i=0}^{12} \sum_{t} m_{t,i} h_{t,i+j} \quad (6)$$

2.2 Second-order Features

One of the contributions of the FANTASTIC toolbox is to include *second-order* features. Second-order features are derivative descriptors that reflect, for a particular feature, how an observed feature value relates to a reference corpus. They help contextualize the values a feature can take. Is this a high number? Is it a common result? Or if the feature is multivariate: is this combination of values typical or atypical, or perhaps representative of a particular style? Examples of second-order features in the FANTASTIC toolbox include features based on document frequencies, i.e. how many songs (documents) in a large corpus contain an observed event or structure: *mctf.mean.log.DF* computes the mean log document frequency over all melodic motives in a given melody.

2.2.1 Second-Order Audio Features in One Dimension

Like many audio features, most of the audio features discussed in this paper are based on frequency-domain computations, which are typically performed on short overlapping windows. As a result, the features discussed here represent continuous-valued, uncountable quantities. Symbolic features, on the other hand, operate on countable collections of events. This makes it impossible to apply the same operations directly to both, and alternatives must be found for the audio domain.

After comparison of several alternatives, we propose a non-parametric measure of typicality based on log odds. The second-order log odds of a feature value x can formally be defined as the *log odds of observing a less extreme value in the reference corpus*. It is conceptually similar to a p -value, which measures the probability of observing a *more* extreme value, but we look at its complement, expressed as odds, and take the log.

We further propose a simple non-parametric approach to compute the above odds. By defining ‘less extreme’ as ‘more probable’, we can make use of density estimation (e.g., kernel density estimation) to obtain a probability density estimate $f(X)$ for the observed feature X , and look at the rank of each feature value’s density in the reference corpus. Normalizing this rank by the number of observations gives us a pragmatic estimate of the probability we’re looking for, and applying the logit function gives us the log odds:

$$Z(X) = \text{logit} \left[\frac{\text{rank}(f(X))}{N} \right] \quad (7)$$

where N is the size of the reference corpus. Since Z is non-parametric and based on ranks, the output always follows the same logistic distribution, which is bell-shaped, symmetric, and general very similar to a normal distribution. The feature can therefore be used out of the box for a variety of statistical applications.

Some caution is warranted when using Z where there are a limited number of observations. If the first order feature X is one-dimensional, some form of density estimation is typically possible even if few data are available. For multivariate features with independent dimensions (e.g.,

MFCC features), each dimension can be treated as a one-dimensional feature, and a meaningful density estimate can also be obtained. However, if the dimensions of a multidimensional feature are not de-correlated by design but highly interdependent (as is the case for chroma features), density estimates require more data. For such cases, a covariance matrix must typically be estimated, increasing the number of parameters to be estimated, and thereby the number of required data points for a fit.

2.2.2 Second-Order Audio Features in d Dimensions

For higher-dimensional features, such as *MIB* and *HIC*, we turn to other measures of typicalness. After comparison of distributions and correlations of several alternatives, we adopt two approaches. The first measure, directly adopted from the FANTASTIC toolbox, is Kendall’s rank-based correlation τ . The second measure is *information* (I), an information-theoretic measure of unexpectedness. This measure assumes that the multidimensional first order feature itself can be seen as a frequency distribution F over possible observations in an audio excerpt (cf. term frequencies), and that a similar distribution F_c can be found for the full reference corpus (cf. document frequencies). We define the $I(F)$ as the average of $-\log F_c$, weighted by F :

$$I(F) = - \sum_{i=1}^d F(i) \log F_c(i) \quad (8)$$

The assumptions hold for HIC, BIM and HI, and produce well-behaved second-order feature values. The result is similar to *mean.log.TFDF*, *mtcf.mean.log.DF* and *mtcf.mean.entropy* in the FANTASTIC toolbox and highly correlated with *mtcf.mean.gl.weight*. Information is also used as a measure of surprise by Pearce [14].

2.3 Song- vs. Corpus-based Second-order Features

In a statistical learning perspective, expectations arise from statistical inference by the listener, who draws on a lifetime of listening experiences to assess whether a particular stimulus is to be expected or not. In [9], Huron compares *veridical* and *schematic* expectations, analogous to episodic and semantic memory. Veridical expectations of a listener are due to familiarity with a specific musical work. Schematic expectations arise from the “auditory generalizations” that help us deal with novel, but broadly familiar situations.

If, in a corpus study, the documents are song segments rather than entire songs, second-order features can be used to incorporate a crude model of both layers of expectation. By choosing the reference corpus to be a collection of fragments spanning a large number of songs, the above measures of typicality and surprise approximate schematic expectations: values that are typical, representative of the reference corpus, are more expected. By choosing as the reference corpus the set of all segments belonging to the same song, veridical expectations can be approximated.

In the following section, we will refer to corpus-based second-order features as *conventionality*. The second, song-based second-order features indicate how representative a

segment is for the song, and to some extent, how much a segment is repeated. We will refer to this as *recurrence*.

3. HOOK DISCOVERY: A CASE STUDY

We tested the proposed approach to audio corpus analysis by examining data from the Hooked! experiment on long-term musical salience [3]. Using these data, we sought to address three questions: (i) how do the proposed audio features behave and what aspects of the music do they model, (ii) which attributes of the music, as measured by both an audio feature set and a selection of symbolic features, predict recognition rating differences within songs, and finally, (iii) how much insight do audio-based corpus analysis tools add when compared to the symbolic feature set?

3.1 Data

The Hooked! experiment used a broad selection of Western pop songs from the 1930s to the present. The experiment tested how quickly and accurately participants could recognise different segments from each song, based on the Echo Nest segmentation algorithm.⁴ For each song segment, the data include an estimate of the *drift rate*, the reciprocal of the amount of time it would take a median participant to recognize the segment, based on linear ballistic accumulation, a cognitive model for timed recognition tasks [2, 4]. To improve reliability, we excluded song segments that fewer than 15 serious participants had attempted to recognize (where a “serious” participant is defined to be a participant who attempted at least 15 segments). We further excluded all segments from songs from which fewer than 3 segments met the previous reliability criteria. After these exclusions, 1715 song segments remained, taken from 321 different songs, representing data from 973 participants. We were unable to obtain symbolic transcriptions of all songs, and so for comparing audio and symbolic features, we used a restricted set of 99 transcribed songs (536 segments).

3.2 Audio Features

For timbre description, we used a feature set that is largely the same as the one used in [18], where statistical analysis of an audio corpus is used to model pop songs choruses. Specifically, we computed the loudness (mean and standard deviations) for each segment, mean sharpness and roughness, and the total variance of the MFCC features. Instead of the pitch centroid feature, we obtained an estimate of pitch height using the *Melodia* melody extraction algorithm and computed the mean.⁵ For chroma, HPCP were used.⁶

For each of these one-dimensional features, we then computed the corpus-based and song-based second-order features as described in Section 2.2.1 using Python.⁷ Finally, we added song and corpus-based $Z(X)$ features based on the mean of the first 13 MFCC components. First-order

⁴ <http://www.echonest.com/>

⁵ <http://mtg.upf.edu/technologies/melodia>

⁶ <http://mtg.upf.edu/technologies/hpcp>

⁷ code will be made available at <http://github.com/jvbalen>

features based on the MFCC means were not included because of their limited interpretability. All features were computed over 15-s segments starting from the beginning of each segment, as participants in the experiment were given a maximum of 15 s for recognition.

For melody and harmony description, we used the features described in Section 2.1, and compute the entropy H as a first-order measure of dispersion. The entropies were then normalized as follows:

$$H' = \log \frac{H_{\max} - H}{H_{\max}} \quad (9)$$

As second-order features, Kendall's τ and the information I were computed, as proposed in Section 2.2.2.

3.3 Symbolic features

The symbolic features used were a subset of 19 first-order and 5 second-order features from the FANTASTIC toolbox, computed for both melodies and bass lines. Second-order features were computed with both the song and the full dataset as a reference, yielding a total of 58 symbolic descriptors.

3.4 Principal Component Analysis

Before going further with either the audio or the symbolic feature sets, we used principal component analysis (PCA) as a way to identify groups of features that may measure a single underlying source of variance and as a way to reduce the dimensionality of the feature spaces to a more manageable number of decorrelated variables. Features were centered and normalized before PCA, and the resulting components were transformed with a varimax rotation to improve interpretability. We selected the number of components to retain (12 in both cases) using parallel analysis [8].

3.5 Linear Mixed Effects Model

In order to fit the extracted components to the drift rates, we used a linear mixed-effects regression model. Mixed-effects models can handle repeated-measures data where several data points are linked to the same song and therefore have a correlated error structure. The Hooked! data provide drift rates for individual sections within songs, and one would indeed expect considerably less variation in drift rates within songs than between them: some pop songs are thought to be much "catchier" than others overall. Moreover, it is likely impossible to model between-song variation in recognizability from content-based features alone: it may arise from differences in marketing, radio play, or social appeal.

Linear mixed-effects models have the further advantage that they are easy to interpret due to the linearity and additivity of the effects of the predictor variables. More complex machine-learning schemes might be able to explain more variance and make more precise predictions for the dependent variable, but this usually comes at the cost of the interpretability of the model.

We fit three models, one including audio components only, one including symbolic components only, and one

including both feature types, and used a stepwise selection procedure at $\alpha = .005$ to identify the most significant predictors under each model. In all models, the dependent variable was the log drift rate of a song segment and the related measures (random effects) were handled as a random intercept, i.e., we added a per-song offset to a traditional linear regression (fixed effects) on song segments, with the assumption that these offsets be distributed normally:

$$\log y_{ij} = \beta' \mathbf{x}_{ij} + u_i + \epsilon_{ij} \quad (10)$$

where i indexes songs, j indexes segments within songs, y_{ij} is the drift rate for song segment ij , \mathbf{x}_{ij} is the vector of standardized feature component scores for song segment ij plus an intercept term, the $u_i \sim N(0, \sigma_{\text{song}}^2)$, and the $\epsilon_{ij} \sim N(0, \sigma_{\text{residual}}^2)$. To facilitate comparison, we fit the audio-only model twice: once using the full set of 321 songs and again using just the 99 songs with transcriptions.

4. RESULTS AND DISCUSSION

4.1 Audio Components

Table 1 displays the component loadings (correlation coefficients between the extracted components and the original features) for audio feature set. The loadings tell a consistent story. The 12 components we retain break the audio feature set down into three timbre components (first order, conventionality, and recurrence) and three entropy components (idem), two features grouping conventionality and recurrence for melody and harmony, respectively, and three more detailed timbre components correlating with sharpness, pitch range and dynamic range.

Component 9 is characterized by an increased dynamic range and MFCC variance and a typical pitch height. We hypothesize that this component correlates with the presence and prominence of vocals. It is not unreasonable to assume that the most typical registers for the melodies in a pop corpus would be the registers of the singing voice, and vocal entries could also be expected to modulate a section's timbre and loudness. This hypothesis is also consistent with our own observations while listening to a selection of fragments at various points along the Component 9 scale.

Overall, the neatness of the above reduction attests to the advantage of using interpretable features, and to the potential of this particular feature set.

4.2 Recognizability Predictors

A look at the first column of results for the linear mixed effects model (Table 2) confirms that the audio features are indeed meaningful descriptors for this corpus. Eight components correlate significantly, most of them relating to conventionality of features. This suggests a general pattern in which more recognizable sections have a more typical, expected sound. Another component, timbral recurrence, points to the role of repetition: sections that are more representative of a song are more recognizable. Finally, the component with the strongest effect is Vocal Prominence.

Feature	Component											
	1	2	3	4	5	6	7	8	9	10	11	12
MIB Song	.31	-.10	.12	.08	.05	.66	.05	.08	.23	.08	-.01	.14
HI Song	-.25	-.08	.12	.06	.11	.55	.12	.35	-.06	.04	.01	-.02
MIB Corpus	.15	-.03	-.02	.13	.00	.77	-.06	.00	.08	-.02	-.01	-.05
HI Corpus	-.28	-.09	-.05	-.01	.10	.55	.11	.42	-.15	-.02	.08	-.05
HIC Song	.04	.13	.22	.04	.00	.13	-.04	.58	-.03	.06	-.02	-.03
HIC Corpus	-.23	.11	.04	.32	.08	.15	-.07	.66	.03	-.06	.07	-.00
HIC Entropy	.88	.06	.03	-.16	.02	.07	-.02	-.23	-.12	.02	-.00	-.10
MIB Entropy	.83	-.15	-.00	-.19	.04	.04	.08	.26	.26	.03	-.02	.20
HI Entropy	.85	-.06	.02	-.20	.01	-.01	.04	.15	.12	.02	-.02	.16
HIC Song Information	.84	.17	.06	.09	.11	.13	-.02	-.16	-.28	-.04	.10	-.13
MIB Song Information	.79	-.21	-.03	.01	.07	.05	.13	.25	.29	.07	-.02	.21
HI Song Information	.90	.18	.01	.11	.07	-.07	.00	-.17	-.03	-.02	.00	-.03
HIC Corpus Information	.86	.16	.06	.01	.10	.11	-.02	-.20	-.27	-.02	.09	-.13
MIB Corpus Information	.79	-.19	-.01	-.03	.07	.02	.14	.26	.31	.07	-.02	.21
HI Corpus Information	.90	.15	.02	-.01	.03	-.12	-.01	-.24	-.03	.00	-.02	-.03
HIB Entropy Song	.03	.11	.42	.08	.03	.00	-.08	.15	.08	.19	.01	-.06
MIB Entropy Song	.01	-.01	.07	.10	.03	-.01	.03	.02	-.01	.82	.00	.05
HI Entropy Song	.03	.02	.11	.12	.06	.04	-.02	-.01	.02	.81	-.01	.02
HIB Entropy Corpus	-.13	.08	.08	.68	.08	.15	-.06	.26	-.03	-.10	.07	-.02
MIB Entropy Corpus	-.04	-.09	-.01	.80	.01	.06	.14	-.01	.05	.16	.00	.07
HI Entropy Corpus	-.03	-.07	-.02	.84	.04	.04	.06	.04	.05	.19	-.02	.04
Loudness	-.04	.92	.07	-.06	-.05	-.05	-.07	.06	-.04	.02	-.07	.04
Roughness	.14	.78	.14	.01	.15	.09	.31	.06	-.08	.07	.06	.01
Melodic Pitch Height	.13	.66	-.05	-.03	.09	-.24	-.16	.09	.22	-.06	-.06	.00
MFCC Variance	.13	-. 51	-.05	.08	-.26	.10	.05	-.02	.48	.02	-.22	-.10
Loudness Song	-.03	-.05	.67	-.01	.06	.01	.07	-.04	.10	.03	.11	-.03
Roughness Song	.04	.10	.67	-.03	-.01	.02	.11	.08	-.05	-.02	-.04	-.05
Mel. Pitch Height Song	-.01	.02	.46	.03	.13	.14	-.12	-.15	.29	.07	.16	.03
MFCC Mean Song	.07	.07	.61	-.04	.21	.12	.10	.10	-.07	.16	.11	.11
MFCC Variance Song	.00	-.04	.54	.03	.01	-.06	.10	.08	-.10	-.09	-.06	.17
Loudness Corpus	.04	-.23	.06	.07	.12	.08	.76	-.05	.22	.02	.10	-.05
Roughness Corpus	.12	.34	.15	.03	.00	.01	.71	-.07	.05	.04	.03	-.07
Mel. Pitch Height Corpus	.00	.04	.06	.06	.25	.06	.14	-.01	.60	.02	.14	-.09
MFCC Mean Corpus	.21	.13	.12	.07	.51	.03	.31	.20	-.18	.05	.14	.08
MFCC Variance Corpus	-.09	-.09	.08	.08	.25	-.02	.40	.05	-.13	-.13	-.12	.21
Sharpness	.23	.11	.03	.08	.72	.04	.29	.13	.08	-.01	.10	.05
Sharpness Song	-.02	-.07	.24	-.04	.50	.06	-.14	-.07	.04	.15	-.08	-.04
Sharpness Corpus	.08	.10	.03	.06	.75	.03	.03	-.02	.14	-.01	-.10	-.01
Loudness SD	.10	.38	.09	.06	-.06	.06	.22	.02	.40	.03	-. 61	-.03
Loudness SD Song	.04	.02	.22	.02	-.05	.00	-.05	.03	.14	.01	.60	.03
Loudness SD Corpus	.03	.05	-.02	.05	-.03	.04	.19	.02	.04	.00	.78	.02
Mel. Pitch SD	.21	-.10	-.02	-.05	.04	-.19	.21	.18	-.27	.12	-.07	-.28
Mel. Pitch SD Song	.01	.04	.11	.01	.04	.13	.00	-.15	.01	.14	.07	.69
Mel. Pitch SD Corpus	.13	.03	-.02	.06	-.01	-.02	.01	.11	-.08	-.04	.00	.74
R^2	.16	.06	.05	.05	.05	.04	.04	.04	.04	.04	.04	.03

Note. MIB = Melodic Interval Bigram; HI = Harmonization Interval; HIC = Harmony Interval Co-occurrence. Loadings > .40 are in boldface. Collectively, these components explain 64 % of the variance in the underlying data. We interpret and name them as follows: (1) Melodic/Harmonic Entropy, (2) Timbral Intensity, (3) Timbral Recurrence, (4) Melodic/Harmonic Entropy Conventionality, (5) Sharpness Conventionality, (6) Melodic Conventionality, (7) Timbral Conventionality, (8) Harmonic Conventionality, (9) Vocal Prominence, (10) Melodic Entropy Recurrence, (11) Dynamic Range Conventionality, and (12) Melodic Range Conventionality.

Table 1. Loadings after varimax rotation for principal component analysis of corpus-based audio features.

Parameter	Audio ^a		Audio ^b		Symbolic ^b		Combined ^b	
	$\hat{\beta}$	99.5 % CI	$\hat{\beta}$	99.5 % CI	$\hat{\beta}$	99.5 % CI	$\hat{\beta}$	99.5 % CI
Fixed effects								
Intercept	-0.84	[-0.91, -0.77]	-0.67	[-0.78, -0.56]	-0.62	[-0.73, -0.51]	-0.63	[-0.74, -0.53]
Audio								
Vocal Prominence	0.14	[0.10, 0.18]	0.11	[0.04, 0.17]			0.08	[0.01, 0.15]
Timbral Conventuality	0.09	[0.05, 0.13]						
Melodic Conventuality	0.06	[0.02, 0.11]						
M/H Entropy Conventuality	0.06	[0.02, 0.10]						
Sharpness Conventuality	0.05	[0.02, 0.09]						
Harmonic Conventuality	0.05	[0.01, 0.10]						
Timbral Recurrence	0.05	[0.02, 0.08]						
Mel. Range Conventuality	0.05	[0.01, 0.08]	0.07	[0.02, 0.13]			0.07	[0.01, 0.12]
Symbolic								
Melodic Repetitivity					0.12	[0.06, 0.19]	0.11	[0.05, 0.17]
Mel./Bass Conventuality					0.07	[0.01, 0.13]	0.08	[0.01, 0.14]
Random effects								
$\hat{\sigma}_{\text{song}}$	0.39	[0.34, 0.45]	0.35	[0.26, 0.45]	0.34	[0.25, 0.44]	0.32	[0.24, 0.42]
$\hat{\sigma}_{\text{residual}}$	0.48	[0.45, 0.50]	0.40	[0.37, 0.44]	0.39	[0.35, 0.43]	0.38	[0.34, 0.42]
$R^2_{\text{marginal}}{}^c$.10		.06		.07		.10	
$R^2_{\text{conditional}}{}^c$.47		.46		.47		.47	
$-2 \times \log \text{likelihood}$	2765.61		699.81		576.74		558.11	

Note. Grouping by song, all models displayed are the optimal random-intercept models for the given feature types after step-wise selection using Satterthwaite-adjusted F -tests at $\alpha = .005$. Component scores – but not log drift rates – were standardized prior to regression.

^a Complete set of 321 songs ($N = 1715$ segments). ^b Reduced set of 99 songs with symbolic transcriptions ($N = 536$ segments).

^c Coefficients of determination following Nakagawa and Schielzeth's technique for mixed-effects models [13]. The marginal coefficient reflects the proportion of variance in the data that is explained by the fixed effects alone and the conditional coefficient the proportion explained by the complete model (fixed and random effects together).

Table 2. Estimated prediction coefficients and variances for audio and symbolic components influencing the relative recognizability (log drift rate) of popular song segments.

The model based on symbolic data only, in the third column, has just two components. This is possibly due to the reduced number of sections available for fitting, as the audio-based model run on the reduced dataset also yields just two components. The top symbolic features that make up the first of the significant components are melodic entropy and productivity, both negatively correlated, suggesting that recognizable melodies are more repetitive. The top features that make up the second components are *mtcf.mean.log.DF*, for the melody (song-based and corpus-based), and negative *mtcf.mean.productivity* (song-based and corpus-based for both bass and melody). This suggests that recognizable melodies contain more typical motives (higher DF, lower second-order productivity).

The last column shows how the combined model, in which both audio and symbolic components were used, retains the same audio and symbolic components that make up the previous two models. The feature sets are, in other words, complementary: not only are all four components still relevant at $\alpha < .005$, the marginal R^2 now reaches .10, as opposed to .06 and .07 for the individual models. This answers the last of the questions stated in Section 3: for the data in this study, the audio-based corpus analysis tools contribute substantial insight, and make an excellent addition to the symbolic feature set.

5. CONCLUSIONS AND FUTURE WORK

We have presented a strategy for audio corpus description that combines a new kind of melodic and harmonic interval profiles, three general-purpose second-order features, and the newly introduced notion of song-based and corpus-based second-order features. Using these features to analyse the results of a hook discovery experiment, we show that all of the above contributions add new and relevant layers of information to the corpus description. We conclude that an audio corpus analysis as proposed in this paper can indeed complement symbolic corpus analysis, which opens a range of opportunities for future work. As possible future directions we would like to perform more experiments on the Hooked! data, exploring more first- and second-order descriptors and more powerful statistical or machine-learning models, to see if allowing for interactions and non-linearities helps to explain more of the variance in drift rates between sections. We also would like to extend the feature set to explore rhythm description and chord estimation, especially as more reliable transcription tools become available from the MIR community.

Acknowledgements

JVB, JAB and DB are supported by COGITCH (NWO CATCH project 640.005.004) and FES project COMMIT/.

6. REFERENCES

- [1] Mathieu Barthet, Mark Plumbley, Alexander Kachkaev, Jason Dykes, Daniel Wolff, and Tillman Weyde. Big chord data extraction and mining. In *Proceedings of the 9th Conference on Interdisciplinary Musicology*, Berlin, Germany, 2014.
- [2] Scott Brown and Andrew Heathcote. The simplest complete model of choice response time: Linear ballistic accumulation. *Cognitive Psychology*, 57(3):153–78, 2008.
- [3] John Ashley Burgoyne, Dimitrios Bountouridis, Jan Van Balen, and Henkjan J. Honing. Hooked: A game for discovering what makes music catchy. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, pages 245–50, Curitiba, Brazil, 2013.
- [4] John Ashley Burgoyne, Jan Van Balen, Dimitrios Bountouridis, Themistoklis Karavellas, Frans Wiering, Remco C. Veltkamp, and Henkjan J. Honing. The contours of catchiness, or Where to look for a hook. Paper presented at the International Conference on Music Perception and Cognition, Seoul, South Korea, 2014.
- [5] John Ashley Burgoyne, Jonathan Wild, and Ichiro Fujinaga. Compositional data analysis of harmonic structures in popular music. In Jonathan Wild, Jason Yust, and John Ashley Burgoyne, editors, *Mathematics and Computation in Music*, pages 52–63. Springer, Berlin, 2013.
- [6] Darrell Conklin and Ian H. Witten. Multiple viewpoint systems for music prediction. *Journal of New Music Research*, 24(1):51–73, 1995.
- [7] Trevor de Clercq and David Temperley. A corpus analysis of rock harmony. *Popular Music*, 30(1):47–70, 2011.
- [8] John L. Horn. A rationale and test for the number of factors in factor analysis. *Psychometrika*, 30(2):179–85, 1965.
- [9] David Huron. *Sweet Anticipation: Music and the Psychology of Expectation*. MIT Press, Cambridge, MA, 2006.
- [10] Matthias Mauch, Robert M MacCallum, Mark Levy, and Armand M Leroi. The evolution of popular music: USA 1960–2010. *Royal Society Open Science*. In press.
- [11] Dirk Moelants, Olmo Cornelis, and Marc Leman. Exploring African tone scales. In *Proceedings of the 10th International Society for Music Information Retrieval Conference*, pages 489–94, Kobe, Japan, 2009.
- [12] Daniel Müllensiefen and Andrea R Halpern. The role of features and context in recognition of novel melodies. *Music Perception: An Interdisciplinary Journal*, 31(5):418–435, 2014.
- [13] Shinichi Nakagawa and Holger Schielzeth. A general and simple method for obtaining R^2 from generalized linear mixed-effects models. *Methods in Ecology and Evolution*, 4(2):133–42, 2013.
- [14] Marcus Thomas Pearce. *The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition*. PhD thesis, City University London, England, 2005.
- [15] Joan Serrà, Alvaro Corral, Marián Bogaña, Martín Haro, and Josep Ll. Arcos. Measuring the evolution of contemporary Western popular music. *Scientific Reports*, 2(521), 2012.
- [16] Xavier Serra, Michela Magas, Emmanouil Benetos, Magdalena Chudy, S. Dixon, Arthur Flexer, Emilia Gómez, F. Gouyon, P. Herrera, Sergi Jordà, Oscar Paytuvi, G. Peeters, Jan Schlüter, H. Vinet, and G. Widmer. *Roadmap for Music Information Research*. MIReS Consortium, 2013.
- [17] Jan Van Balen, Dimitrios Bountouridis, Frans Wiering, and Remco Veltkamp. Cognition-inspired descriptors for scalable cover song retrieval. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, pages 379–384, Taipei, Taiwan, 2014.
- [18] Jan Van Balen, John Ashley Burgoyne, Frans Wiering, and Remco C. Veltkamp. An analysis of chorus features in popular song. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil, 2013.

LARGE-SCALE CONTENT-BASED MATCHING OF MIDI AND AUDIO FILES

Colin Raffel, Daniel P. W. Ellis
LabROSA, Department of Electrical Engineering
Columbia University, New York, NY
{craffel, dpwe}@ee.columbia.edu

ABSTRACT

MIDI files, when paired with corresponding audio recordings, can be used as ground truth for many music information retrieval tasks. We present a system which can efficiently match and align MIDI files to entries in a large corpus of audio content based solely on content, i.e., without using any metadata. The core of our approach is a convolutional network-based cross-modality hashing scheme which transforms feature matrices into sequences of vectors in a common Hamming space. Once represented in this way, we can efficiently perform large-scale dynamic time warping searches to match MIDI data to audio recordings. We evaluate our approach on the task of matching a huge corpus of MIDI files to the Million Song Dataset.

1. TRAINING DATA FOR MIR

Central to the task of content-based Music Information Retrieval (MIR) is the curation of ground-truth data for tasks of interest (e.g. timestamped chord labels for automatic chord estimation, beat positions for beat tracking, prominent melody time series for melody extraction, etc.). The quantity and quality of this ground-truth is often instrumental in the success of MIR systems which utilize it as training data. Creating appropriate labels for a recording of a given song by hand typically requires person-hours on the order of the duration of the data, and so training data availability is a frequent bottleneck in content-based MIR tasks.

MIDI files that are time-aligned to matching audio can provide ground-truth information [8, 25] and can be utilized in score-informed source separation systems [9, 10]. A MIDI file can serve as a timed sequence of note annotations (a “piano roll”). It is much easier to estimate information such as beat locations, chord labels, or predominant melody from these representations than from an audio signal. A number of tools have been developed for inferring this kind of information from MIDI files [6, 7, 17, 19].

Haleyv et al. [11] argue that some of the biggest successes in machine learning came about because “...a large training set of the input-output behavior that we seek to automate is available to us in the wild.” The motivation behind

```
J/Jerseygi.mid  
V/VARIA180.MID  
Carpenters/WeveOnly.mid  
2009 MIDI/handy_man1-D105.mid  
G/Garotos Modernos - Bailanta De Fronteira.mid  
Various Artists/REWINDNAS.MID  
GoldenEarring/Twilight_Zone.mid  
Sure.Polyphone.Midi/Poly 2268.mid  
d/danza3.mid  
100sure.polyphone_midi/Fresh.mid  
rogers_kenny/medley.mid  
2009 MIDI/looking_out_my_backdoor3-Bb192.mid
```

Figure 1. Random sampling of 12 MIDI filenames and their parent directories from our corpus of 455,333 MIDI files scraped from the Internet.

this project is that MIDI files fit this description. Through a large-scale web scrape, we obtained 455,333 MIDI files, 140,910 of which were unique – orders of magnitude larger than any available dataset of aligned transcriptions. This proliferation of data is likely due to the fact that MIDI files are typically a few kilobytes in size and were therefore a popular format for distributing and storing music recordings when hard drives had only megabytes of storage.

The mere existence of a large collection of MIDI data is not enough: In order to use MIDI files as ground truth, they need to be both matched (paired with a corresponding audio recording) and aligned (adjusted so that the timing of the events in the file match the audio). Alignment has been studied extensively [8, 25], but prior work typically assumes that the MIDI and audio have been correctly matched. Given large corpora of audio and MIDI files, the task of matching entries of each type may seem to be a simple matter of fuzzy text matching of the files’ metadata. However, MIDI files almost never contain structured metadata, and as a result the best-case scenario is that the artist and song title are included in the file or directory name. While we found some examples of this in our collection of scraped MIDI files, the vast majority of the files had effectively no metadata information. Figure 1 shows a random sampling of directory and filenames from our collection.

Since the goal of matching MIDI and audio files is to find pairs that have *content* in common, we can in principle identify matches regardless of metadata availability or accuracy. However, comparing content is more complicated and more expensive than a fuzzy text match. Since NM comparisons are required to match a MIDI dataset of size N to an audio file dataset of size M , matching large collections



© Colin Raffel, Daniel P. W. Ellis.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Colin Raffel, Daniel P. W. Ellis. “Large-Scale Content-Based Matching of MIDI and Audio Files”. 16th International Society for Music Information Retrieval Conference, 2015.

is practical only when the individual comparisons can be made very fast. Thus, the key aspect of our work is a *highly-efficient* scheme to match the content of MIDI and audio files. Our system learns a cross-modality hashing which converts both MIDI and audio content vectors to a common Hamming (binary) space in which the “local match” operation at the core of dynamic time warping (DTW) reduces to a very fast table lookup. As described below, this allows us to match a single MIDI file to a huge collection of audio files in minutes rather than hours.

The idea of using DTW distance to match MIDI files to audio recordings is not new. For example, in [13], MIDI-audio matching is done by finding the minimal DTW distance between all pairs of chromagrams of (synthesized) MIDI and audio files. Our approach differs in a few key ways: First, instead of using chromagrams (a hand-designed representation), we learn a common representation for MIDI and audio data. Second, our datasets are many orders of magnitude larger (hundreds of thousands vs. hundreds of files), which necessitates a much more efficient approach. Specifically, by mapping to a Hamming space we greatly speed up distance matrix calculation and we receive quadratic speed gains by implicitly downsampling the audio and MIDI feature sequences as part of our learned feature mapping.

In the following section, we detail the dataset of MIDI files we scraped from the Internet and describe how we prepared a subset for training our hasher. Our cross-modality hashing model is described in Section 3. Finally, in section 4 we evaluate our system’s performance on the task of matching files from our MIDI dataset to entries in the Million Song Dataset [3].

2. PREPARING DATA

Our project began with a large-scale scrape of MIDI files from the Internet. We obtained 455,333 files, of which 140,910 were found to have unique MD5 checksums. The great majority of these files had little or no metadata information. The goal of the present work is to develop an efficient way to match this corpus against the Million Song Dataset (MSD), or, more specifically, to the short preview audio recordings provided by 7digital [20].

For evaluation, we need a collection of ground-truth MIDI-audio pairs which are correctly matched. Our approach can then be judged based on how accurately it is able to recover these pairings using the content of the audio and MIDI files alone. To develop our cross-modality hashing scheme, we further require a collection of *aligned* MIDI and audio files, to supply the matching pairs of feature vectors from each domain that will be used to train our model for hashing MIDI and audio features to a common Hamming space (described in Section 3). Given matched audio and MIDI files, existing alignment techniques can be used to create this training data; however, we must exclude incorrect matches and failed alignments. Even at the scale of this reduced set of training data, manual alignment verification is impractical, so we developed an improved alignment quality score which we describe in Section 2.3.

2.1 Metadata matching

To obtain a collection of MIDI-audio pairs, we first separated a subset of MIDI files for which the directory name corresponded to the song’s artist and the filename gave the song’s title. The resulting metadata needed additional canonicalization; for example, “The Beatles”, “Beatles, The”, “Beatles”, and “The Beatles John Paul Ringo George” all appeared as artists. To normalize these issues, we applied some manual text processing and resolved the artists and song titles against the Freebase [5] and Echo Nest¹ databases. This resulted in a collection of 17,243 MIDI files for 10,060 unique songs, which we will refer to as the “clean MIDI subset”.

We will leverage the clean MIDI subset in two ways: First, to obtain ground-truth pairings of MSD/MIDI matches, and second, to create training data for our hashing scheme. The training data does not need to be restricted to the MSD, and using other sources to increase the training set size will likely improve our hashing performance, so we combined the MSD with three benchmark audio collections: CAL500 [26], CAL10k [24], and uspop2002 [2]. To match these datasets to the clean MIDI subset, we used the Python search engine library `whoosh`² to perform a fuzzy matching of their metadata. This resulted in 26,311 audio/MIDI file pairs corresponding to 5,243 unique songs.

2.2 Aligning audio to synthesized MIDI

Fuzzy metadata matching is not enough to ensure that we have MIDI and audio files with matching content: For instance, the metadata could be incorrect, the fuzzy text match could have failed, the MIDI could be a poor transcription (e.g., missing instruments or sections), and/or the MIDI and audio data could correspond to different versions of the song. Since we will use DTW to align the audio content to an audio resynthesis of the MIDI content [8, 13, 25], we could potentially use the overall match cost – the quantity minimized by DTW – as an indicator of valid matches, since unrelated MIDI and audio pairs will likely result in a high optimal match cost. (An overview of DTW and its application to music can be found in [18].)

Unfortunately, the calibration of this raw match cost “confidence score” is typically not comparable between different alignments. Our application, however, requires a DTW confidence score that can reliably decide when an audio/MIDI file pairing is valid for use as training data for our hashing model. Our best results came from the following system for aligning a single MIDI/audio file pair: First, we synthesize the MIDI data using `fluidsynth`.³ We then estimate the MIDI beat locations using the MIDI file’s tempo change information and the method described in [19]. To circumvent the common issue where the beat is tracked one-half beat out of phase, we double the BPM until it is at least 240. We compute⁴ beat locations for the audio signal with the constraint that the BPM should remain close to the

¹ <http://developer.echonest.com/docs/v4>

² <https://github.com/dokipen/whoosh>

³ <http://www.fluidsynth.org>

⁴ All audio analysis was accomplished with `librosa` [16].

global MIDI tempo. We then compute log-amplitude beat-synchronous constant-Q transforms (CQTs) of audio and synthesized MIDI data with semitone frequency spacing and a frequency range from C3 (65.4 Hz) to C7 (1046.5 Hz). The resulting feature matrices are then of dimensionality $N \times D$ and $M \times D$ where N and M are the resulting number of beats in the MIDI and audio recordings respectively and D is 48 (the number of semitones between C3 and C7). Example CQTs computed from a 7digital preview clip and from a synthesized MIDI file can be seen in Figure 2(a) and 2(b) respectively.

We then use DTW to find the lowest-cost path through a full pairwise cosine distance matrix $S \in \mathbb{R}^{N \times M}$ of the MIDI and audio CQTs. This path can be represented as two sequences $p, q \in \mathbb{R}^L$ of indices from each sequence such that $p[i] = n, q[i] = m$ implies that the n th MIDI beat should be aligned to the m th audio beat. Traditional DTW constrains this path to include the start and end of each sequence, i.e. $p[1] = q[1] = 1$ and $p[L] = N; q[L] = M$. However, the MSD audio consists of cropped preview clips from 7digital, while MIDI files are generally transcriptions of the entire song. We therefore modify this constraint so that either $gN \leq p[L] \leq N$ or $gM \leq q[L] \leq M$; g is a parameter which provides a small amount of additional tolerance and is normally close to 1. We employ an additive penalty ϕ for “non-diagonal moves” (i.e. path entries where either $p[i] = p[i + 1]$ or $q[i] = q[i + 1]$) which, in our setting, is set to approximate a typical distance value in S . The combined use of g and ϕ typically results in paths where both $p[1]$ and $q[1]$ are close to 1, so no further path constraints are needed. For synthesized MIDI-to-audio alignment, we used $g = .95$ and set ϕ to the 90th percentile of all the values in S . The cosine distance matrix and the lowest-cost DTW path for the CQTs shown in Figure 2(a) and 2(b) can be seen in Figure 2(c).

2.3 DTW cost as confidence score

The cost of a DTW path p, q through S is calculated by the sum of the distances between the aligned entries of each sequence:

$$c = \sum_{i=1}^L S[p[i], q[i]] + T(p[i] - p[i - 1], q[i] - q[i - 1])$$

where the transition cost term $T(u, v) = 0$ if u and v are 1, otherwise $T(u, v) = \phi$. As discussed in [13], this cost is not comparable between different alignments for two main reasons: Firstly, the path length can vary greatly across MIDI/audio file pairs depending on N and M . We therefore prefer a per-step *mean* distance, where we divide c by L . Secondly, various factors irrelevant to alignment such as differences in production and instrumentation can effect a global shift on the values of S , even when its local variations still reveal the correct alignment. This can be mitigated by normalizing the DTW cost by the mean value of the submatrix of S containing the DTW path:

$$B = \sum_{i=\min(p)}^{\max(p)} \sum_{j=\min(q)}^{\max(q)} S[i, j]$$

We combine the above to obtain a modified DTW cost \hat{c} :

$$\hat{c} = \frac{c}{LB}$$

To estimate the largest value of \hat{c} for acceptable alignments, we manually auditioned 125 alignments and recorded whether the audio and MIDI were well-synchronized for their entire duration, our criterion for acceptance. This ground-truth supported a receiver operating characteristic (ROC) for \hat{c} with an AUC score of 0.986, indicating a highly reliable confidence metric. A threshold of 0.78 allowed zero false accepts on this set while only falsely discarding 15 well-aligned pairs. Retaining all alignments with costs better (lower) than this threshold resulted in 10,035 successful alignments.

Recall that these matched and aligned pairs serve two purposes: They provide training data for our hashing model; and we also use them to evaluate the entire content-based matching system. For a fair evaluation, we exclude items used in training from the evaluation, thus we split the successful alignments into three parts: 50% to use as training data, 25% as a “development set” to tune the content-based matching system, and the remaining 25% to use for final evaluation of our system. Care was taken to split based on *songs*, rather than by entry (since some songs appear multiple times).

3. CROSS-MODALITY HASHING OF MIDI AND AUDIO DATA

We now arrive at the central part of our work, the scheme for hashing both audio and MIDI data to a common simple representation to allow very fast computation of the distance matrix S needed for DTW alignment. In principle, given the confidence score of the previous section, to find audio content that matches a given MIDI file, all we need to do is perform alignment against every possible candidate audio file and choose the audio file with the lowest score. To maximize the chances of finding a match, we need to use a large and comprehensive pool of audio files. We use the 994,960 7digital preview clips corresponding to the Million Song Dataset, which consist of (typically) 30 second portions of recordings from the largest standard research corpus of popular music [20]. A complete search for matches could thus involve 994,960 alignments for each of our 140,910 MIDI files.

The CQT-to-CQT approach of section 2.2 cannot feasibly achieve this. The median number of beats in our MIDI files is 1218, and for the 7digital preview clips it is 186. Computing the cosine distance matrix S of this size (for $D = 48$ dimension CQT features) using the highly optimized C++ code from *scipy* [14] takes on average 9.82 milliseconds on an Intel Core i7-4930k processor. When implemented using the LLVM just-in-time compiler Python module *numba*,⁵ the DTW cost calculation described above takes on average 892 microseconds on the same processor. Matching a *single* MIDI file to the MSD using this approach would thus take just under three hours;

⁵<http://numba.pydata.org/>

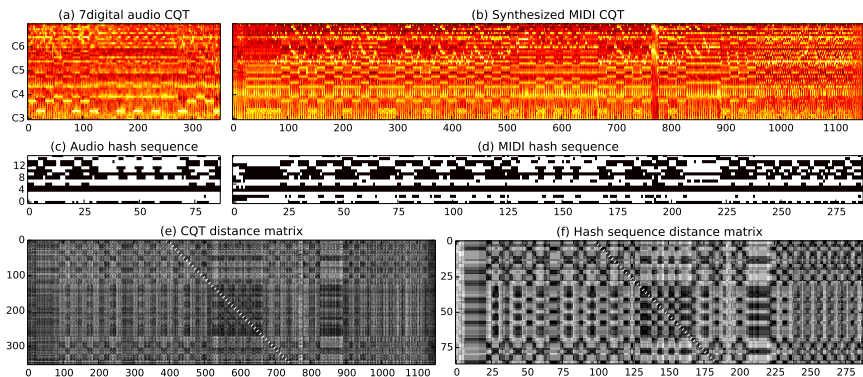


Figure 2. Audio and hash-based features and alignment for Billy Idol - “Dancing With Myself” (MSD track ID TRCZQLG128F427296A). (a) Normalized constant-Q transform of 7digital preview clip, with semitones on the vertical axis and beats on the horizontal axis. (b) Normalized CQT for synthesized MIDI file. (c) Hash bitvector sequence for 7digital preview clip, with pooled beat indices and Hamming space dimension on the horizontal and vertical axes respectively. (d) Hash sequence for synthesized MIDI. (e) Distance matrix and DTW path (displayed as a white dotted line) for CQTs. Darker cells indicate smaller distances. (f) Distance matrix and DTW path for hash sequences.

matching our entire 140,910 MIDI file collection to the MSD would take years. Clearly, a more efficient approach is necessary.

Calculating the distance matrix and the DTW cost are both $\mathcal{O}(NM)$ in complexity; the distance matrix calculation is about 10 times slower presumably because it involves D multiply-accumulate operations to compute the inner product for each point. Calculating the distance between feature vectors is therefore the bottleneck in our system, so any reduction in the number of feature vectors (i.e., beats) in each sequence will give quadratic speed gains for both DTW and distance matrix calculations.

Motivated by these issues, we propose a system which learns a common, reduced representation for the audio and MIDI features in a Hamming space. By replacing constant-Q spectra with bitvectors, we replace the expensive inner product computation by an exclusive-or operation followed by simple table lookup: The exclusive-or of two bitvectors a and b will yield a bitvector consisting of 1s where a and b differ and 0s elsewhere, and the number of 1s in all bitvectors of length D can be precomputed and stored in a table of size 2^D . In the course of computing our Hamming space representation, we also implicitly downsample the sequences over time, which provides speedups for both distance matrix and DTW calculation. Our approach has the additional potential benefit of *learning* the most effective representation for comparing audio and MIDI constant-Q spectra, rather than assuming the cosine distance of CQT vectors is suitable.

3.1 Hashing with convolutional networks

Our hashing model is based on the Siamese network architecture proposed in [15]. Given feature vectors $\{x\}$ and $\{y\}$ from two modalities, and a set of pairs \mathcal{P} such that $(x, y) \in \mathcal{P}$ indicates that x and y are considered “similar”, and a second set \mathcal{N} consisting of “dissimilar” pairs, a nonlinear mapping is learned from each modality to a common Hamming space such that similar and dissimilar feature vectors are respectively mapped to bitvectors with small and large Hamming distances. A straightforward objective function which can be minimized to find an appropriate mapping is

$$\mathcal{L} = \frac{1}{|\mathcal{P}|} \sum_{(x,y) \in \mathcal{P}} \|f(x) - g(y)\|_2^2 - \frac{\alpha}{|\mathcal{N}|} \sum_{(x,y) \in \mathcal{N}} \max(0, m - \|f(x) - g(y)\|_2)^2$$

where f and g are the nonlinear mappings for each modality, α is a parameter to control the importance of separating dissimilar items, and m is a target separation of dissimilar pairs.

The task is then to optimize the nonlinear mappings f and g with respect to \mathcal{L} . In [15] the mappings are implemented as multilayer nonlinear networks. In the present work, we will use convolutional networks due to their ability to exploit invariances in the input feature representation; CQTs contain invariances in both the time and frequency axes, so convolutional networks are particularly well-suited for our task. Our two feature modalities are CQTs from synthesized MIDI files and audio files. We assemble the set of “similar” cross-modality pairs \mathcal{P} by taking the CQT

frames from individual aligned beats in our training set. The choice of \mathcal{N} is less obvious, but randomly choosing CQT spectra from non-aligned beats in our collection achieved satisfactory results.

3.2 System specifics

Training the hashing model involves presenting training examples and backpropagating the gradient of \mathcal{L} through the model parameters. We held out 10% of the training set described in Section 2 as a validation set, not used in training the networks. We z-scored the remaining 90% across feature dimensions and re-used the means and standard deviations from this set to z-score the validation set.

For efficiency, we used minibatches of training examples; each minibatch consisted of 50 sequences obtained by choosing a random offset for each training sequence pair and cropping out the next 100 beats. For \mathcal{N} , we simply presented the network with subsequences chosen at random from different songs. Each time the network had iterated over minibatches from the entire training set (one epoch), we repeated the random sampling process. For optimization, we used RMSProp, a recently-proposed stochastic optimization technique [23]. After each 100 minibatches, we computed the loss \mathcal{L} on the validation set. If the validation loss was less than 99% of the previous lowest, we trained for 1000 more iterations (minibatches).

While the validation loss is a reasonable indicator of network performance, its scale will vary depending on the α and m regularization hyperparameters. To obtain a more consistent metric, we also computed the distribution of distances between the hash vectors produced by the network for the pairs in \mathcal{P} and those in \mathcal{N} . To directly measure network performance, we used the Bhattacharyya distance [4] to compute the separation of these distributions.

In each modality, the hashing networks have the same architecture: A series of alternating convolutional and pooling layers followed by a series of fully-connected layers. All layers except the last use rectifier nonlinearities; as in [15], the output layer uses a hyperbolic tangent. This choice allows us to obtain binary hash vectors by testing whether each output unit is greater or less than zero. We chose 16 bits for our Hamming space, since 16 bit values are efficiently manipulated as short unsigned integers. The first convolutional layer has 16 filters each of size 5 beats by 12 semitones, which gives our network some temporal context and octave invariance. As advocated by [21], all subsequent convolutional layers had $2^{n+3} 3 \times 3$ filters, where n is the depth of the layer. All pooling layers performed max-pooling, with a pooling size of 2×2 . Finally, as suggested in [12], we initialized all weights with normally-distributed random variables with mean of zero and a standard deviation of $\sqrt{2/n_{in}}$, where n_{in} is the number of inputs to each layer. Our model was implemented using `theano` [1] and `lasagne`.⁶

To ensure good performance, we optimized all model hyperparameters using Whetlab,⁷ a web API which im-

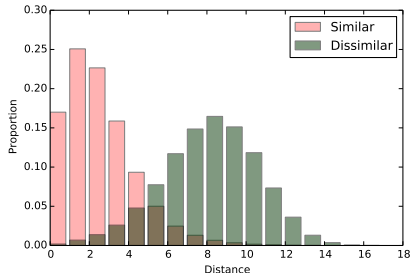


Figure 3. Output hash distance distributions for our best-performing network.

plements black-box Bayesian optimization [22]. We used Whetlab to optimize the number of convolutional/pooling layers, the number and size of the fully-connected layers, the RMSProp learning rate and decay parameters, and the α and m regularization parameters of \mathcal{L} . As a hyperparameter optimization objective, we used the Bhattacharyya distance as described above. The best performing network found by Whetlab had 2 convolutional layers, 2 “hidden” fully-connected layers with 2048 units in addition to a fully-connected output layer, a learning rate of .001 with an RMSProp decay parameter of .65, $\alpha = .5$, and $m = 4$. This hyperparameter configuration yielded the output hash distance distributions for \mathcal{P} and \mathcal{N} shown in Figure 3, for a Bhattacharyya separation of 0.488.

4. MATCHING MIDI FILES TO THE MSD

After training our hashing system as described above, the process of matching MIDI collections to the MSD proceeds as follows: First, we precompute hash sequences for every 7digital preview clip and every MIDI file in the clean MIDI subset. Note that in this setting we are not computing feature sequences for known MIDI/audio pairs, so we cannot force the audio’s beat tracking tempo to be the same as the MIDI’s; instead, we estimate their tempos independently. Then, we compute the DTW cost as described in Section 2.2 between every audio and MIDI hash sequence.

We tuned the parameters of the DTW cost calculation to optimize results over our “development” set of successfully aligned MIDI/MSD pairs. We found it beneficial to use a smaller value of $g = 0.9$. Using a fixed value for the non-diagonal move penalty avoids the percentile calculation, so we chose $\phi = 4$. Finally, we found that normalizing by the average distance value \mathcal{B} did not help, so we skipped this step.

4.1 Results

Bitvector sequences for the CQTs shown in Figure 2(a) and 2(b) can be seen in 2(c) and 2(d) respectively. Note

it would be ending its service. For posterity, the results of our hyperparameter search are available at <http://bit.ly/hash-param-search>.

⁶ <https://github.com/Lasagne/Lasagne>

⁷ Between submission and acceptance of this paper, Whetlab announced

Rank	1	10	100	1000	10000
Percent \leq	15.2	41.6	62.8	82.7	95.9

Table 1. Percentage of MIDI-MSD pairs whose hash sequences had a rank better than each threshold.

that because our networks contain two downsample-by-2 pooling layers, the number of bitvectors is $\frac{1}{4}$ of the number of constant-Q spectra for each sequence. The Hamming distance matrix and lowest-cost DTW path for the hash sequences are shown in Figure 2(f). In this example, we see the same structure as in the CQT-based cosine distance matrix of 2(e), and the same DTW path was successfully obtained.

To evaluate our system using the known MIDI-audio pairs of our evaluation set, we rank MSD entries according to their hash sequence DTW distance to a given MIDI file, and determine the rank of the correct match for each MIDI file. The correct item received a mean reciprocal rank of 0.241, indicating that the correct matches tended to be ranked highly. Some intuition about the system performance is given by reporting the percentage of MIDI files in the test set where the correct match ranked below a certain threshold; this is shown for various thresholds in Table 1.

Studying Table 1 reveals that we can't rely on the correct entry appearing as the top match among all the MSD tracks; the DTW distance for true matches only appears at rank 1 about 15.2% of time. Furthermore, for a significant portion of our MIDI files, the correct match did not rank in the top 1000. This was usually caused by the MIDI file being beat tracked at a different tempo than the audio file, which inflated the DTW score. For MIDI files where the true match ranked highly but not first, the top rank was often a different version (cover, remix, etc.) of the correct entry. Finally, some degree of inaccuracy can be attributed to the fact that our hashing model is not perfect (as shown in Figure 3) and that the MSD is very large, containing many possible decoys. In a relatively small proportion of cases, the MIDI hash sequence ended up being very similar to many MSD hash sequences, pushing down the rank of the correct entry.

Given that we cannot reliably assume the top hit from hashing is the correct MSD entry, it is more realistic to look at our system as a pruning technique; that is, it can be used to discard MSD entries which we can be reasonably confident do not match a given MIDI file. For example, Table 1 tells us that we can use our system to compute the hash-based DTW score between a MIDI file and every entry in the MSD, then discard all but 1% of the MSD and only risk discarding the correct match about 4.1% of the time. We could then perform the more precise DTW on the full CQT representations to find the best match in the remaining candidates. Pruning methods are valuable only when they are substantially faster than performing the original computation; fortunately, our approach is orders of magnitude faster: On the same Intel Core i7-4930k processor, for the median hash sequence lengths, calculating a Hamming

distance matrix between hash sequences is about 400 times faster than computing the CQT cosine distance matrix (24.8 microseconds vs. 9.82 milliseconds) and computing the DTW score is about 9 times faster (106 microseconds vs. 892 microseconds). These speedups can be attributed to the fact that computing a table lookup is much more efficient than computing the cosine distance between two vectors and that, thanks to downsampling, our hash-based distance matrices have $\frac{1}{16}$ of the entries of the CQT-based ones. In summary, a straightforward way to describe the success of our system is to observe that we can, with high confidence, discard 99% of the entries of the MSD by performing a calculation that takes about as much time as matching against 1% of the MSD.

5. FUTURE WORK

Despite our system's efficiency, we estimate that performing a full match of our 140,910 MIDI file collection against the MSD would still take a few weeks, assuming we are parallelizing the process on the 12-thread Intel i7-4930k. There is therefore room for improving the efficiency of our technique. One possibility would be to utilize some of the many pruning techniques which have been proposed for the general case of large-scale DTW search. Unfortunately, most of these techniques rely on the assumption that the query sequence is of the same length or shorter than all the sequences in the database and so would need to be modified before being applied to our problem. In terms of accuracy, as noted above most of our hash-match failures can be attributed to erroneous beat tracking. With a better beat tracking system or with added robustness to this kind of error, we could improve the pruning ability of our approach. We could also compare the accuracy of our system to a slower approach on a much smaller task to help pinpoint failure modes. Even without these improvements, our proposed system will successfully provide orders of magnitude of speedup for our problem of resolving our huge MIDI collection against the MSD. All the code used in this project is available online.⁸

6. ACKNOWLEDGEMENTS

We would like to thank Zhengshan Shi and Hilary Mogul for preliminary work on this project and Eric J. Humphrey and Brian McFee for fruitful discussions.

7. REFERENCES

- [1] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian J. Goodfellow, Arnaud Bergeron, Nicolas Bouchard, and Yoshua Bengio. Theano: new features and speed improvements. Deep Learning and Unsupervised Feature Learning NIPS Workshop, 2012.
- [2] Adam Berenzweig, Beth Logan, Daniel P. W. Ellis, and Brian Whitman. A large-scale evaluation of acoustic

⁸ <http://github.com/craffel/midi-dataset>

- and subjective music-similarity measures. *Computer Music Journal*, 28(2):63–76, 2004.
- [3] Thierry Bertin-Mahieux, Daniel P. W. Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 591–596, 2011.
- [4] Anil Kumar Bhattacharyya. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society*, 35:99–109, 1943.
- [5] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250. ACM, 2008.
- [6] Michael Scott Cuthbert and Christopher Ariza. `music21`: A toolkit for computer-aided musicology and symbolic music data. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pages 637–642, 2010.
- [7] Tuomas Eerola and Petri Toivainen. MIR in Matlab: The MIDI toolbox. In *Proceedings of the 5th International Society for Music Information Retrieval Conference*, pages 22–27, 2004.
- [8] Sebastian Ewert, Meinard Müller, Verena Konz, Daniel Müllensiefen, and Geraint A. Wiggins. Towards cross-version harmonic analysis of music. *IEEE Transactions on Multimedia*, 14(3):770–782, 2012.
- [9] Sebastian Ewert, Bryan Pardo, Mathias Muller, and Mark D. Plumbley. Score-informed source separation for musical audio recordings: An overview. *IEEE Signal Processing Magazine*, 31(3):116–124, 2014.
- [10] Joachim Ganseman, Gautham J. Mysore, Paul Scheunders, and Jonathan S. Abel. Source separation by score synthesis. In *Proceedings of the International Computer Music Conference*, pages 462–465, 2010.
- [11] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009.
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. *arXiv preprint arXiv:1502.01852*, 2015.
- [13] Ning Hu, Roger B. Dannenberg, and George Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 185–188, 2003.
- [14] Eric Jones, Travis Oliphant, and Pearu Peterson. SciPy: Open source scientific tools for Python. 2014.
- [15] Jonathan Masci, Michael M. Bronstein, Alexander M. Bronstein, and Jürgen Schmidhuber. Multimodal similarity-preserving hashing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(4):824–830, 2014.
- [16] Brian McFee, Matt McVicar, Colin Raffel, Dawen Liang, and Douglas Repetto. `librosa`: v0.3.1, November 2014.
- [17] Cory McKay and Ichiro Fujinaga. `jSymbolic`: A feature extractor for MIDI files. In *Proceedings of the International Computer Music Conference*, pages 302–305, 2006.
- [18] Meinard Müller. *Information retrieval for music and motion*. Springer, 2007.
- [19] Colin Raffel and Daniel P. W. Ellis. Intuitive analysis, creation and manipulation of MIDI data with `pretty_midi`. In *Proceedings of the 15th International Society for Music Information Retrieval Conference Late Breaking and Demo Papers*, 2014.
- [20] Alexander Schindler, Rudolf Mayer, and Andreas Rauber. Facilitating comprehensive benchmarking experiments on the million song dataset. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, pages 469–474, 2012.
- [21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [22] Jasper Snoek, Hugo Larochelle, and Ryan P. Adams. Practical bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems*, pages 2951–2959, 2012.
- [23] Tijmen Tieleman and Geoffrey Hinton. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*, 4, 2012.
- [24] Derek Tingle, Youngmoo E. Kim, and Douglas Turnbull. Exploring automatic music annotation with “acoustically-objective” tags. In *Proceedings of the international conference on Multimedia information retrieval*, pages 55–62. ACM, 2010.
- [25] Robert J. Turetsky and Daniel P. W. Ellis. Ground-truth transcriptions of real music from force-aligned MIDI syntheses. *Proceedings of the 4th International Society for Music Information Retrieval Conference*, pages 135–141, 2003.
- [26] Douglas Turnbull, Luke Barrington, David Torres, and Gert Lanckriet. Towards musical query-by-semantic-description using the CAL500 data set. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 439–446. ACM, 2007.

IMPROVING GENRE ANNOTATIONS FOR THE MILLION SONG DATASET

Hendrik Schreiber
tagtraum industries incorporated
hs@tagtraum.com

ABSTRACT

Any automatic music genre recognition (MGR) system must show its value in tests against a ground truth dataset. Recently, the public dataset most often used for this purpose has been proven problematic, because of mislabeling, duplications, and its relatively small size. Another dataset, the Million Song Dataset (MSD), a collection of features and metadata for one million tracks, unfortunately does not contain readily accessible genre labels. Therefore, multiple attempts have been made to add song-level genre annotations, which are required for supervised machine learning tasks. Thus far, the quality of these annotations has not been evaluated.

In this paper we present a method for creating additional genre annotations for the MSD from databases, which contain multiple, crowd-sourced genre labels per song (Last.fm, beaTunes). Based on label co-occurrence rates, we derive taxonomies, which allow inference of top-level genres. These are most often used in MGR systems.

We then combine multiple datasets using majority voting. This both promises a more reliable ground truth and allows the evaluation of the newly generated and pre-existing datasets. To facilitate further research, all derived genre annotations are publicly available on our website.

1. INTRODUCTION

Automatic music genre recognition (MGR) is among the most popular Music Information Retrieval (MIR) tasks [5]. Until 2012, the majority of datasets used for MGR research was private and the most popular public dataset was GTZAN [13, 14]. Unfortunately, GTZAN has some documented deficiencies [12]. Additionally, with 1,000 excerpts from ten different genres, GTZAN is relatively small by today’s standards. Desirable as dataset for MGR, in terms of size and available features, is the Million Song Dataset (MSD) [2]. But by 2012, when it was still very new, only three of the 345 publications (0.7%) surveyed in [13] had used it. This may be explained by the fact that the MSD does not contain explicit genre annotations. The authors of all three publications first had to derive

song-level genre labels for a subset of the MSD as ground truth. For this purpose, Hu [7] and Schindler [10] both used album-level genre labels scraped from the All Music Guide website¹. Dieleman et al. [3] selected 20 commonly used genres from the MusicBrainz artist tags contained in the MSD—an approach similar to what the MSD author suggested for the MSD Genre Dataset, a “simplified genre dataset from the Million Song Dataset for teaching purposes”². With the exception of [10], the used ground truths aren’t re-usable or well documented. And in the case of [10], they have not been evaluated and don’t allow for multiple genre annotations per song.

In the spirit of [9], what is required to help facilitate MGR research using the MSD, is a song-level ground truth with a documented level of accuracy that also allows for ambiguity. In the following sections we will first derive (where necessary) and then compare four different genre datasets for the MSD. In Section 2 we describe how we created the beaTunes Genre Dataset (BDG). In Section 3, we apply a similar approach to the Last.fm Dataset³, creating a Last.fm Genre Dataset (LFMGD). In Section 4 we explore to which degree the BDG, LFMGD and the datasets created by Hu (HO)⁴ and Schindler (Top-MAGD) agree, and derive two new datasets, by combining multiple sources. Finally, in Section 5 and Section 6, we define benchmark partitions to promote repeatability of experiments using the new datasets and point to additional raw data.

2. BEATUNES GENRE DATASET

beaTunes⁵ is a consumer application that encourages its users to correct song metadata using multiple heuristics. It also supports sending anonymized metadata to a central database, which matches it to metadata sent by other users. Much like tags on Last.fm, this allows keeping track of multiple user-submitted genres per song. For example, one song may have been associated with the label `ROCK` by five users, while three users regarded the same song a `POP` song. The database currently contains more than 870



© Hendrik Schreiber.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Hendrik Schreiber. “Improving genre annotations for the million song dataset”, 16th International Society for Music Information Retrieval Conference, 2015.

¹ <http://allmusic.com/>

² <http://labrosa.ee.columbia.edu/millionsong/blog/11-2-28-deriving-genre-dataset>

³ <http://labrosa.ee.columbia.edu/millionsong/lastfm>

⁴ <http://web.cs.miami.edu/home/yajiehu/resource/genre/>

⁵ <http://www.beatunes.com/>

million user song submissions of which 772 million are labeled with a genre and mapped to more than 85 million songs. Furthermore, the database stores each user's system language. In the remainder of this section we describe, how we used the existing genre labels to assign top-level genres (seeds) to each song and matched them to songs in the MSD.

2.1 Genre Label Normalization

In the beaTunes database, more than one million different, user-submitted genre labels are stored. Some of these are slight spelling variations of popular genre names like Hip-Hop or composites of multiple genres like Hip-Hop/Rap. Others describe custom categorization schemes, ratings, or are simply noise. In order to extract the most-used and thus most important genre labels from the database, we first normalized their names and then ranked them by usage count. The following normalization procedure was employed (building on [6]):

1. Convert to lowercase
2. Remove whitespace
3. Convert 'n', and, and & in different spellings of R&B, D&B, and Rock'n'Roll to n
4. Replace alt. with alternative and trad. with traditional
5. Tokenize with +&/, ; : ! \ [] () as delimiters
6. From each token, remove all characters that aren't letters or digits
7. Sort tokens alphanumerically
8. Concatenate tokens with / as delimiter

This effectively treats composite labels like Hip-Hop/Rap as their own genre, but makes sure that Hip-Hop/Rap is equal to Rap/Hip-Hop. The special treatment in step 3 for R&B, D&B, and Rock'n'Roll is necessary, as the & character is also used as delimiter in composite labels (e.g. Christian & Gospel). After normalization, almost 700,000 different genre labels remain. However, 50% of all user-submitted songs are covered by the 16 most-used genres, 80% by the top 131 genres, and 90% by the top 750 genres.

2.2 Language-Specific Counts

Since genre labels reflect how listeners with a specific cultural background perceive music and what it means to them [1, 4, 8], we investigated how the collection's top genre rankings differ when taking the user's system language into account. Not surprisingly, by and large they are quite similar—with Rock, Pop, Hip-Hop and Jazz occurring in most top tens (Table 1). But there are a few notable exceptions. English speaking listeners are the only ones with Country (ranked 9th) in their top ten genres, French speakers rank Reggae (5th) higher than others, Spanish speakers rank Latin (5th), House (7th), and Otros (8th) high, and Japanese speakers rank J-Pop (3rd) near the top. Clearly, these differences are indicative of cultural preferences and should be taken into account when creating genre taxonomies. Therefore, in the

remainder of this paper, we have only used the beaTunes label-submissions of English-speaking users.

2.3 Inferring Genre Taxonomies

As the beaTunes database contains on average about nine user submissions (i.e. genre labels) per song, we can record co-occurrences of labels on a per-song basis and thus infer relationships between them. Latent Semantic Analysis (LSA) with cosine similarity has been used for this purpose before [11]. But because we did not plan on using the cosine distance as metric, we did not deem it necessary to use Singular Value Decomposition (SVD) to keep the dimensionality low. Instead, we opted for a much simpler method. We filtered out rarely used labels and restricted ourselves to the top 1,000 genres covering over 93% of all user submissions with genre information.

Formally, we define $G := \{\text{Rock}, \text{Pop}, \dots\}$ with $|G| = 1000 = n$ as the set of the n top genres, which are stored as distinct values in the vector $g \in G^n$ with $g := (\text{Rock}, \text{Pop}, \dots)$. Each user submission is defined as a sparse vector $u \in \mathbb{N}^n$ with

$$u_i = \begin{cases} 1, & \text{if } g_i = \text{user-submitted genre} \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

To establish the connection between a song s and its user labels u , we simply add up all u 's belonging to the song and divide by the number of u 's. Thus each song is represented by a vector $s \in \mathbb{R}^n$ with $0 \leq s_i \leq 1$ and $\sum_{i=0}^{n-1} s_i = 1$, denoting each genre's relative strength. To compute the co-occurrences for a given genre g_i with all other genres g , we element-wise average all s for which $s_i \neq 0$ is true. I.e.:

$$C_{g_i} := \bar{s}, \quad \text{for all } s \text{ with } s_i \neq 0; C \in \mathbb{R}^{n \times n} \quad (2)$$

The result is the matrix C that allows us to see how often a given genre co-occurs with another genre. Note that C is not symmetric as it would have been, had we used SVD with cosine similarity. So just because Alternative co-occurs with Rock fairly strongly ($C_{\text{Alternative}, \text{Rock}} = 0.156$), the opposite is not necessarily true ($C_{\text{Rock}, \text{Alternative}} = 0.026$, see Table 2). From the C values for the beaTunes database, it is also obvious, that Rock and Pop can be distinguished very well—both labels co-occur much more with themselves than with the other (Rock: 0.609/0.057, Pop: 0.593/0.077).

We exploit the asymmetry of C to construct a taxonomy by defining the following two rules:

(1) If a genre a co-occurs with another genre b more than a minimum threshold τ , and a co-occurs with b more than the other way around, then we assume that a is a sub-genre of b . More formally:

$$\begin{aligned} & a \text{ is a sub-genre of } b, \text{ if} \\ & a \neq b \\ & \wedge C_{a,b} > \tau \\ & \wedge C_{b,a} > C_{a,b} \\ & \text{for all } a, b \in G \end{aligned} \quad (3)$$

	All ($N = 772.1$)	English ($N = 521.1$)	German ($N = 97.9$)	French ($N = 43.3$)	Spanish ($N = 27.1$)	Japanese ($N = 11.0$)
1.	Rock	Rock	Pop	Rock	Rock	Rock
2.	Pop	Pop	Rock	Pop	Pop	Pop
3.	Alternative	Alternative	Electronic	Jazz	Jazz	J-Pop
4.	Jazz	Hip-Hop/Rap	Hip-Hop	Hip-Hop	Soundtrack	R&B
5.	Hip-Hop	Hip-Hop	Jazz	Reggae	Latin	Soundtrack
6.	Hip-Hop/Rap	R&B	Alternative	R&B	Dance	Jazz
7.	Soundtrack	Soundtrack	Dance	Soundtrack	House	Electronica/Dance
8.	R&B	Jazz	R&B	Blues	Otros	ロック (Rock)
9.	Electronic	Country	Rock/Pop	Electronic	Blues	Altern. & Punk
10.	Country	Altern. & Punk	Soundtrack	Rap	Electronica	Hip-Hop/Rap

Table 1. Top ten genres used by beaTunes users with different languages. N denotes the number of submissions in millions.

Co-Occurrence Rank	1.	2.	3.	4.
Rock	Rock (0.609)	Pop (0.057)	Alternative (0.026)	Rock/Pop (0.016)
Pop	Pop (0.593)	Rock (0.077)	Rock/Pop (0.014)	R&B (0.013)
Alternative	Alternative (0.394)	Rock (0.156)	Pop (0.052)	Alternative/Punk (0.036)
R&B	R&B (0.566)	Pop (0.061)	Soul (0.036)	R&B/Soul (0.033)
Soundtrack	Soundtrack (0.754)	Rock (0.024)	Pop (0.022)	Game (0.011)
...

Table 2. Genre labels in the beaTunes database and their top four co-occurring labels ordered by relative strength given in parenthesis. The underlying values from the co-occurrence matrix C were computed taking only submissions by English speakers and the 1,000 most-used labels into account.

(2) Because this rule allows a genre to be a sub-genre of multiple genres, we add:

$$\begin{aligned}
 &a \text{ is a } \textit{direct} \text{ sub-genre of } b, \text{ iff} \\
 &a \text{ is a sub-genre of } b \\
 \wedge &C_{a,b} > C_{a,c} \\
 &\text{with } c \neq a \wedge c \neq b; a, b, c \in G
 \end{aligned}
 \tag{4}$$

By finding all direct sub-genres and their parents, we can now create a set of trees. The number of created trees depends on the threshold τ . We found, that to properly distinguish between genres like Pop, Rock, Dance, R&B, Folk, and Other, $\tau := 0.085$ proved to be useful, resulting in 141 trees. The roots of these trees are typically the names of seed-genres like Jazz, Pop, Rock, etc. (see Figure 1).

Not all generated trees have children. For example, the tree with the seed-genre Groove consists of just the root. Although Groove co-occurs with R&B, Rock, Funk, and Soul, the co-occurrence rates with genres other than itself are all below τ . Even the co-occurrence with itself is low (0.157). This suggests, that Groove is not really a genre, but more a property of a genre. Another example for a root-only tree is Calypso. Here the co-occurrence with itself is much higher (0.606) and indeed Calypso qualifies as stand-alone genre that simply does not have any sub-genres in this database.

Naturally, the generated taxonomies are only simplified mappings of the more complex relationship graph represented by C . In reality, genres aren't necessarily exclusive members of one tree or another (e.g. fusion genres). An ontology is the much better construct. But, as we will see, for the purpose of mapping most sub-genres to their seed-genre, trees are useful.

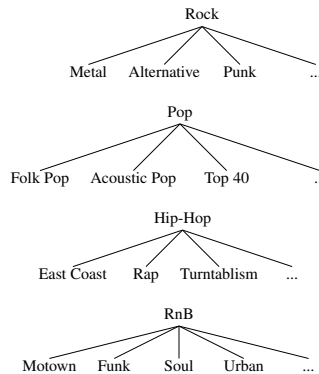


Figure 1. Partial, generated trees for the seed-genres Rock, Pop, Hip-Hop, and R&B.

2.4 Matching with Million Song Dataset

To create song-level genre annotations for the MSD, we queried the beaTunes database for songs with artist/title pairs contained in the MSD and were able to match 677,038 songs. In order to ease the comparison with the HO and Top-MAGD datasets, we associated each matched song with the seed-genre of its most often occurring genre label, taking advantage of the taxonomies created in Section 2.3. Motown, for example, is represented by its seed-genre RnB. In many cases, the found seed-genres are

Co-Occurrence Rank	1.	2.	3.	4.
rock	rock (0.128)	alternative (0.023)	pop (0.021)	indie (0.021)
pop	pop (0.107)	rock (0.037)	femalevocalists (0.024)	80s (0.018)
alternative	alternative (0.076)	rock (0.062)	indie (0.037)	alternativerock (0.023)
indie	indie (0.108)	rock (0.045)	alternative (0.034)	indierock (0.026)
electronic	electronic (0.119)	dance (0.026)	trance (0.021)	electronica (0.019)
...

Table 3. Tags in the Last.fm dataset and their top four co-occurring labels ordered by relative strength given in parenthesis, based on the co-occurrence matrix C , computed taking the 1,000 most-used labels into account.

equal to the All Music Guide labels used by Top-MAGD (Blues, Country, Electronic, International, Jazz, Latin, Pop_Rock, Rap, Reggae, RnB, New Age, Folk, Vocal). With a few exceptions: In our generated taxonomy for English users, Blues and Vocal are not seed-genres, but rather sub-genres of Rock and Jazz, respectively. Therefore, in these cases we used the label itself instead. We also translated World to International, and Pop, Rock, and Pop/Rock to Pop_Rock, and Hip-Hop to Rap. All songs we could not map to a Top-MAGD label were dropped, leaving us with 609,865 songs—90% of the originally matched songs. We call this dataset the beaTunes Genre Dataset (BGD).

3. LAST.FM GENRE DATASET

The Last.fm dataset is similar to the beaTunes database, in that it also contains multiple user-submitted labels per song which are each associated with a weight. Therefore we can use the same method to build a co-occurrence matrix and construct genre trees. The main difference lies in the kind of labels used. While the beaTunes labels are almost exclusively genre names, Last.fm tags vary a lot in content. Many are also genre labels, but others describe a mood, situation, location, time, or something completely different. As the dataset contains 522,366 different tags, it is not feasible to manually extract only the genre related ones. Therefore we again chose to incorporate the 1,000 most-used tags into computed genre trees. Because a single Last.fm song is often associated with many more tags than a beaTunes song with genre labels, we had to choose a different τ . Just like for BGD, we wanted to be able to see genres like Electronic, Jazz, Pop and Rock as seed-genres and therefore chose $\tau := 0.040$, which allows for this (see Table 3 for sample co-occurrence values).

To create the Last.fm Genre Dataset (LFMGD), we associated each song with the seed-genre of the strongest tag that has a seed-genre corresponding to a Top-MAGD label or already corresponds to one of the Top-MAGD labels itself. In either case, we adjusted the spelling suitably. We also translated hiphop to Rap, and pop, rock, poprock to Pop_Rock, and world to International. Again, all songs not easily mappable to a Top-MAGD label were removed from the set. This left us with 340,323 (67.4%) of the 505,216 tracks originally labeled with at least one tag.

	Top-MAGD	LFMGD	BGD
HO	56.6%	52.7%	54.9%
Top-MAGD	-	75.8%	84.1%
LFMGD	-	-	81.0%

Table 5. Pairwise agreement rates for all four datasets for 136,639 MSD tracks occurring in all sets. The highest agreement is set in **bold**, the lowest in *italic*.

Dataset	Top-MAGD	LFMGD	BGD
Agreement Rate	90.4%	87.2%	95.8%

Table 6. Agreement rates for genre labels in Top-MAGD, LFMGD, and BGD when compared with the 133,676 tracks in CD1, found by majority voting.

4. CONSTRUCTING GROUND TRUTH

To construct a reliable ground truth, we evaluated agreement rates between the existing and constructed datasets using the genre labels from Top-MAGD. We then combined the more promising sets (Section 4.1). Because Top-MAGD labels as the lowest common denominator are somewhat unsatisfying, we then used just LFMGD and BGD to construct an additional dataset with finer genre granularity (Section 4.2).

4.1 Truth by Majority

After removal of duplicates⁶, we found 136,639 tracks occurring in all four datasets Top-MAGD, LFMGD, BGD, and HO, all labeled with Top-MAGD genres. As a relative measure of trustworthiness, we calculated their pairwise agreement rate (Table 5). While the rates between Top-MAGD, LFMGD, and BGD are above 75%, those involving HO are below 57%. Unlike the other sets, HO was created with a combined classifier and is not the result of crowd-sourcing or any kind of expert annotation. Therefore a lower agreement rate was to be expected. The almost 20 percentage points difference illustrates that HO is not suitable as ground truth.

Since the other datasets were in relatively high agreement and we did not have a strong reason to believe, that

⁶ <http://labrosa.ee.columbia.edu/millionsong/blog/11-3-15-921810-song-dataset-duplicates>

Top-MAGD	Blues	Country	Electronic	Folk	International	Jazz	Latin	New Age	Pop.Rock	Rap	Reggae	RnB	Vocal
Blues	78.1%	0.1%	0.0%	0.4%	0.1%	1.5%	0.0%	17.9%	0.0%	0.0%	0.0%	1.7%	0.0%
Country	0.0%	86.1%	0.0%	2.3%	0.1%	0.2%	0.0%	0.0%	11.4%	0.0%	0.0%	0.0%	0.0%
Electronic	0.0%	0.0%	82.4%	0.0%	0.4%	0.2%	0.1%	0.1%	15.7%	0.0%	0.0%	0.2%	0.0%
Folk	0.0%	0.8%	0.1%	49.2%	14.9%	0.0%	0.2%	0.1%	34.3%	0.0%	0.0%	0.0%	0.3%
International	0.1%	0.0%	7.8%	0.3%	83.6%	0.7%	0.8%	1.2%	4.5%	0.0%	0.0%	0.4%	0.7%
Jazz	0.1%	0.0%	2.2%	0.0%	0.5%	76.2%	1.2%	0.8%	6.5%	0.2%	0.0%	1.5%	10.8%
Latin	0.0%	0.0%	0.3%	0.0%	0.7%	0.3%	95.6%	0.0%	2.5%	0.2%	0.0%	0.0%	0.3%
New Age	0.0%	0.0%	2.7%	0.0%	1.4%	0.9%	0.0%	93.5%	1.5%	0.0%	0.0%	0.0%	0.0%
Pop.Rock	0.0%	0.1%	1.0%	0.2%	0.6%	0.1%	0.9%	0.0%	96.0%	1.3%	0.0%	0.8%	0.2%
Rap	0.0%	0.0%	3.0%	0.0%	0.7%	0.0%	0.2%	0.0%	4.5%	91.0%	0.0%	0.4%	0.0%
Reggae	0.0%	0.0%	2.2%	0.0%	1.7%	0.1%	1.0%	0.0%	21.5%	0.2%	72.2%	0.1%	0.0%
RnB	0.0%	0.0%	0.5%	0.0%	0.0%	0.0%	0.0%	0.0%	3.3%	0.4%	0.0%	95.8%	0.0%
Vocal	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%	2.9%	0.0%	1.3%	0.0%	0.0%	0.0%	95.8%

BGD	Blues	Country	Electronic	Folk	International	Jazz	Latin	New Age	Pop.Rock	Rap	Reggae	RnB	Vocal
Blues	97.6%	0.0%	0.0%	0.2%	0.0%	0.1%	0.0%	0.0%	1.4%	0.0%	0.0%	0.6%	0.0%
Country	0.1%	97.8%	0.0%	0.4%	0.0%	0.0%	0.0%	0.0%	1.6%	0.0%	0.0%	0.1%	0.0%
Electronic	0.2%	0.0%	91.2%	0.0%	0.4%	0.3%	0.0%	0.6%	6.5%	0.6%	0.1%	0.2%	0.0%
Folk	0.4%	0.8%	0.0%	93.9%	0.2%	0.0%	0.1%	0.1%	3.6%	0.0%	0.0%	0.0%	0.0%
International	0.0%	0.2%	0.7%	0.9%	93.8%	0.5%	0.7%	0.5%	2.2%	0.0%	0.5%	0.0%	0.0%
Jazz	0.1%	0.0%	0.4%	0.0%	0.1%	97.5%	0.2%	0.1%	1.1%	0.1%	0.0%	0.4%	0.0%
Latin	0.1%	0.0%	0.5%	0.3%	1.6%	0.7%	91.3%	0.0%	4.9%	0.2%	0.3%	0.1%	0.0%
New Age	0.1%	0.0%	0.6%	0.1%	0.9%	0.4%	0.0%	97.4%	0.6%	0.0%	0.0%	0.0%	0.0%
Pop.Rock	0.3%	0.3%	1.3%	0.6%	0.1%	0.2%	0.0%	96.4%	0.1%	0.1%	0.3%	0.0%	0.0%
Rap	0.1%	0.0%	0.9%	0.0%	0.0%	0.1%	0.0%	0.0%	4.5%	96.5%	0.2%	0.8%	0.0%
Reggae	0.2%	0.0%	0.3%	0.0%	0.2%	0.0%	0.0%	0.0%	0.4%	0.5%	98.3%	0.1%	0.0%
RnB	0.1%	0.0%	0.2%	0.0%	0.1%	0.2%	0.0%	0.0%	3.8%	0.6%	0.0%	94.9%	0.0%
Vocal	1.3%	0.0%	0.0%	0.0%	0.4%	16.3%	0.4%	0.0%	16.3%	0.0%	0.0%	0.4%	64.9%

LFMGD	Blues	Country	Electronic	Folk	International	Jazz	Latin	New Age	Pop.Rock	Rap	Reggae	RnB	Vocal
Blues	92.3%	0.4%	0.2%	0.2%	0.0%	1.2%	0.0%	0.0%	5.3%	0.1%	0.2%	0.0%	0.0%
Country	0.2%	91.8%	0.0%	1.2%	0.0%	0.2%	0.0%	0.0%	6.4%	0.1%	0.1%	0.0%	0.0%
Electronic	0.0%	0.3%	85.0%	0.1%	0.2%	2.7%	0.1%	0.2%	3.8%	1.1%	0.2%	0.1%	0.0%
Folk	1.3%	3.7%	0.0%	87.6%	0.1%	0.6%	0.0%	0.0%	6.3%	0.0%	0.1%	0.0%	0.2%
International	0.1%	0.2%	2.4%	17.2%	64.7%	3.0%	1.4%	1.2%	7.9%	0.3%	1.4%	0.0%	0.3%
Jazz	0.4%	0.1%	0.5%	0.0%	0.3%	95.1%	0.1%	0.1%	2.8%	0.1%	0.1%	0.0%	0.3%
Latin	0.2%	0.2%	1.6%	1.2%	2.4%	3.8%	59.0%	0.1%	29.6%	0.6%	0.9%	0.0%	0.3%
New Age	0.1%	0.1%	12.1%	2.9%	1.4%	17.6%	0.9%	54.8%	9.8%	0.1%	0.0%	0.0%	0.1%
Pop.Rock	1.2%	1.1%	3.4%	2.8%	0.1%	1.0%	0.1%	0.1%	88.9%	0.4%	0.7%	0.1%	0.2%
Rap	0.0%	0.1%	1.4%	0.0%	0.0%	1.2%	0.3%	0.0%	4.0%	92.2%	0.3%	0.4%	0.0%
Reggae	0.0%	0.0%	0.2%	0.0%	0.1%	0.2%	0.1%	0.0%	2.2%	0.3%	96.6%	0.2%	0.0%
RnB	3.2%	0.2%	0.5%	0.1%	0.0%	9.1%	0.1%	0.0%	20.5%	3.9%	0.4%	61.2%	0.7%
Vocal	0.4%	1.7%	0.4%	0.8%	2.1%	16.7%	1.3%	1.3%	25.9%	2.1%	0.0%	0.0%	47.3%

Table 4. Confusion matrices between CD1 and Top-MAGD, BGD, and LFMGD. Values greater 10% are set in bold.

one of them is better than the other, we constructed a Combined Dataset 1 (CD1) from them using unweighted majority voting. CD1 contains only those tracks, that are labeled exclusively with the Top-MAGD genre set and for which the majority of labels from Top-MAGD, LFMGD, and BGD are identical. MSD duplicates were removed. Out of 136,991 tracks we found a majority genre for 133,676 (97.6% of all), of which 98,149 were found by unanimous consent (73.4% of majorities). To document ambiguity, we recorded both the majority decision and the minority vote, if there was one. This may be used in the evaluation of MGR systems, e.g. for fractional scores, or as indicator for uncertainty. The majority genre distribution of CD1 is shown in Figure 2. *Rock_Pop* is with 59.8% by far the most dominant genre, *Vocal* with 0.2% the most under-represented one.

When comparing Top-MAGD, LFMGD, and BGD to the majority labels from CD1, we found that BGD matches best with 95.8%, followed by Top-MAGD with 90.4%, and LFMGD with 87.2% (Table 6). We believe that the relatively low agreement rate for LFMGD indicates room for improvement in the used mapping procedure from tags to genres, rather than problems with the original *Last.fm* dataset. Even though Top-MAGD was derived from album-level genre labels, it agrees with CD1 remarkably well, which attests to the quality of the set. BGD might be seen as the best of both worlds: its data source

is song-level like LFMGD and at the same time somewhat limited to a genre vocabulary—more like Top-MAGD than LFMGD. This means the problematic mapping from free-form tags to genres is much easier. Overall, one might interpret these numbers as estimates for an upper boundary of MGR systems that test against a ground truth with only one genre label per song.

To provide more detail regarding the individual weaknesses of the datasets relative to CD1, we created confusion matrices (Table 4). In Top-MAGD the largest misclassifications occur for Folk (34.3%), Reggae (21.5%), Blues (17.9%), Electronic (15.7%), and Country (11.4%), which are all categorized as *Pop.Rock*. BGD classifies *Vocal* relatively poorly: 16.3% are misclassified as *Jazz* and 16.3% as *Pop.Rock*. LFMGD tends to misclassify *Latin*, *RnB*, and *Vocal* as *Pop.Rock* (29.6%, 20.5%, 25.9%), and *Vocal* as *Jazz* (16.7%). In summary, most errors occur with songs falsely identified as *Rock_Pop*. Additionally, *Vocal* tends to be misclassified as *Jazz*. We suspect this happens mainly, because *Vocal* is not seen as a genre, but rather as a style.

4.2 Truth by Consensus

Similar to Top-MAGD, almost 60% of all songs in CD1 are labeled *Pop.Rock*. Obviously, this rather coarse labeling is unsatisfying. Therefore we decided to create another

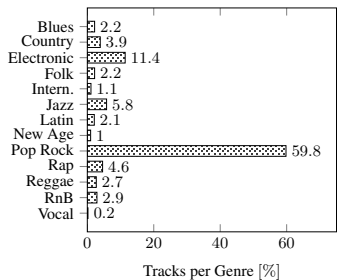


Figure 2. Majority genre distribution of tracks in CD1.

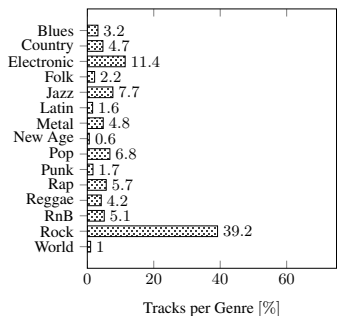


Figure 3. Genre distribution of tracks in CD2C.

dataset, Combined Dataset 2 (CD2), which differentiates between these two genres and adds two additional ones that are popular among users of *beaTunes* and *Last.fm* (*Metal* and *Punk*). Because *International* is hardly used in user-submitted tags and thus seems artificial, we used *World* instead. We also translated *Soul* to *R&B* in order to group them together, and removed *Vocal*, because it is the genre BGD and LFMGD confused most in CD1.

As sources for CD2 we used suitably modified versions of LFMGD and BGD and found 280,831 songs that both fit our genre-set, occur in both datasets, and aren't duplicates. 191,401 (68.2%) of the songs in CD2 have only one genre label, found by consensus. For convenience, we created another dataset called Combined Dataset 2 Consensus (CD2C) containing just those songs. As shown in Figure 3, the genre distribution for CD2C is a little more even than CD1—*Rock* being represented with a 39.2% share, *Pop* with 6.8%, and *New Age* with 0.6%.

5. BENCHMARK PARTITIONS

Inspired by [10] we provide three kinds of benchmark partitions for CD1, CD2, and CD2C in order to promote repeatability of experiments beyond *x*-fold cross validation. These partitions are:

- “Traditional” splits into training and test sets, with sizes 90%, 80%, 66%, and 50%; no stratification.
- Splits into training and test sets, with sizes 90%, 80%, 66%, and 50% and genre stratification.
- Splits with a fixed number of training samples per genre (1,000/2,000/3,000). Genres with fewer songs than the training size were dropped.

As CD2 songs are not always labeled with a majority genre, we used the first listed genre for stratification.

6. ADDITIONAL DATA

BGD and LFMGD represent simplified views on reality, suitable for comparisons with other, similar datasets like *Top-MAGD*. They both assign only one genre per song and the genre labels themselves are very limited. Both simplifications are problematic [9], which is why the combined datasets presented in this paper contain multiple genre labels where feasible. But for both BGD and LFMGD there is actually much more information available on a per-song basis. We are publishing it on our website in the hope that it proves useful for further research. Specifically, this includes:

- Multiple genre annotations/tags per song along with relative strength, and number of user-submissions to judge reliability.
- Co-occurrence matrices computed as described in Section 2.3.
- Derived genre taxonomies.

All data can be found at http://www.tagtraum.com/msd_genre_datasets.html.

7. CONCLUSION AND FUTURE WORK

Reliable and accessible annotations for large datasets are an important precondition for the development of successful music genre recognition (MGR) systems. Some often-used reference datasets are either relatively small or suffer from other deficiencies. To promote the adoption of the Million Song Dataset (MSD) for MGR research, we both evaluated existing and created two new genre annotation datasets for subsets of the MSD. Given that the large sizes of the datasets render manual validation almost impossible, we used either majority voting or consensus to validate existing data, and allowed for ambiguity in the created ground truths. In direct comparison with the generated ground truth CD1, 90.4% of the compared *Top-MAGD* labels were in agreement. To further promote experimentation and comparability, we also provided traditional and stratified benchmark partitions, as well as most of the data the combined datasets were derived from. In the process of creating the new datasets, we used simplifications like English-only labels and trees instead of graphs. Future work is needed to overcome these simplifications and better model the real world.

We hope the provided datasets prove useful for future publications in order to create better MGR systems.

8. REFERENCES

- [1] Jean-Julien Aucouturier and Francois Pachet. Representing musical genre: A state of the art. *Journal of New Music Research*, 32(1):83–93, 2003.
- [2] Thierry Bertin-Mahieux, Daniel PW Ellis, Brian Whitman, and Paul Lamere. The million song dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, pages 591–596, 2011.
- [3] Sander Dieleman, Philémon Brakel, and Benjamin Schrauwen. Audio-based music classification with a pretrained convolutional network. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR)*, pages 669–674, 2011.
- [4] Franco Fabbri. A theory of musical genres: Two applications. *Popular Music Perspectives*, pages 52–81, 1981.
- [5] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. A survey of audio-based music classification and annotation. *Multimedia, IEEE Transactions on*, 13(2):303–319, 2011.
- [6] Gijs Geleijnse, Markus Schedl, and Peter Knees. The quest for ground truth in musical artist tagging in the social web era. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, pages 525–530, 2007.
- [7] Yajie Hu and Mitsunori Oghihara. Genre classification for million song dataset using confidence-based classifiers combination. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1083–1084. ACM, 2012.
- [8] Jin Ha Lee, Kahyun Choi, Xiao Hu, and J Stephen Downie. K-pop genres: A cross-cultural exploration. In *Proceedings of the 14th International Conference on Music Information Retrieval (ISMIR)*, pages 529–534, 2013.
- [9] Cory McKay and Ichiro Fujinaga. Musical genre classification: Is it worth pursuing and how can it be improved? In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, pages 101–106, 2006.
- [10] Alexander Schindler, Rudolf Mayer, and Andreas Rauber. Facilitating comprehensive benchmarking experiments on the million song dataset. In *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR)*, pages 469–474, 2012.
- [11] Mohamed Sordo, Oscar Celma, Martin Blech, and Enric Guaus. The quest for musical genres: Do the experts and the wisdom of crowds agree? In *Proceedings of the 9th International Conference on Music Information Retrieval (ISMIR)*, pages 255–260, 2008.
- [12] Bob L Sturm. An analysis of the gtzan music genre dataset. In *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, pages 7–12. ACM, 2012.
- [13] Bob L Sturm. A survey of evaluation in music genre recognition. In *Adaptive Multimedia Retrieval: Semantics, Context, and Adaptation*, pages 29–66. Springer, 2014.
- [14] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.

A SOFTWARE FRAMEWORK FOR MUSICAL DATA AUGMENTATION

Brian McFee^{1,2,*}, Eric J. Humphrey^{2,3}, and Juan P. Bello²

¹Center for Data Science, New York University

²Music and Audio Research Laboratory, New York University

³MuseAmi, Inc.

ABSTRACT

Predictive models for music annotation tasks are practically limited by a paucity of well-annotated training data. In the broader context of large-scale machine learning, the concept of “data augmentation” — supplementing a training set with carefully perturbed samples — has emerged as an important component of robust systems. In this work, we develop a general software framework for augmenting annotated musical datasets, which will allow practitioners to easily expand training sets with musically motivated perturbations of both audio and annotations. As a proof of concept, we investigate the effects of data augmentation on the task of recognizing instruments in mixed signals.

1. INTRODUCTION

Musical audio signals contain a wealth of rich, complex, and highly structured information. The primary goal of content-based music information retrieval (MIR) is to analyze, extract, and summarize music recordings in a human-friendly format, such as semantic tags, chord and melody annotations, or structural boundary estimations. Modeling the vast complexity of musical audio seems to require large, flexible models with many parameters. By the same token, parameter estimation in large models often requires a large number of samples: big models require big data.

Within the past few years, this phenomenon of increasing model complexity has been observed in the computer vision literature. Currently, the best-performing models for recognition of objects in images exploit two fundamental properties to overcome the difficulty of fitting large, complex models: access to large quantities of annotated data, and label-invariant data transformations [14]. The benefits of large training collections are obvious, but unfortunately difficult to achieve for most musical annotation tasks due to the complexity of the label space and need for expert annotators. However, the idea of generating perturbations of a training set — known as *data augmentation* — can be readily adapted to musical tasks.

*Please direct correspondence to brian.mcfee@nyu.edu



© Brian McFee, Eric J. Humphrey, Juan P. Bello.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Brian McFee, Eric J. Humphrey, Juan P. Bello. “A software framework for musical data augmentation”, 16th International Society for Music Information Retrieval Conference, 2015.

Conceptually, data augmentation consists of the application of one or more deformations to a collection of (annotated) training samples. Data augmentation is motivated by the observation that a learning algorithm may generalize better if it is trained on instances which have been perturbed in ways which are irrelevant to their labels. Some concrete examples of deformations drawn from computer vision include translation, rotation, reflection, and scaling. These simple operations are appealing because they typically do not affect the target class label: an image of a cat still contains a cat, even when it is flipped upside-down.

More generally, deformations apply not only to observable features, but the labels as well. Continuing with the image example, if an image is rotated, then any pixel-wise label annotations (e.g., bounding boxes) should be rotated accordingly. This observation opens up several interesting possibilities for musical applications, in which the target concept space typically exhibits a high degree of structure. A musical analog to the image rotation example would be time-stretching, where time-keyed annotation boundaries (e.g., chord labels or instrument activations) must be adjusted to fit the stretched signal [16].

Many natural, musically-inspired deformations would not only change the position of annotations, but the *values* themselves. For instance, if a time-stretched track has tempo annotations, the annotation values should be scaled accordingly. Similarly, pitch-shifting a track should induce transpositions of annotated fundamental frequency curves, and if the transposition is sufficiently large, chord labels or symbolic annotations may change as well. Because the annotation spaces for music tasks often exhibit a high degree of structure, successful application of data augmentation may require a more sophisticated approach in MIR than in other domains.

1.1 Our contributions

In this work, we describe the MUDA software architecture for applying data augmentation to music information retrieval tasks.¹ The system is designed to be simple, modular, and extensible. The design enables practitioners to develop custom deformations, and combine multiple simple deformations together into pipelines which can generate large volumes of reliably deformed, annotated music data. The proposed system is built on top of JAMS [12],

¹<https://bmcfee.github.io/muda>

which provides a simple container for accessing and transporting multiple annotations for a given track.

We demonstrate the proposed data augmentation architecture with the application of recognizing instruments in mixed signals, and show that simple manipulations can yield improvements in accuracy.

2. RELATED WORK

The first step in developing a solution to an MIR problem is often to design features which discard information thought to be irrelevant to the target concept. For example, chroma features are designed to capture pitch class information and suppress timbre, loudness, or octave height [18]. Similarly, many authors interested in modeling timbre use Mel-frequency cepstral coefficients (MFCCs) and discard the first component to achieve invariance to loudness [19]. This general strategy makes intuitive sense, but it carries many limitations. First, it is not necessarily easy to identify all relevant symmetries in the data: if it was, the modeling problem would be essentially solved. Second, even if such properties are easy to identify, it may still be difficult to engineer appropriately invariant features without discarding potentially useful information. For example, 2-D Fourier magnitude coefficients achieve invariance to time- and pitch-transposition, but discard phase coherence [8].

As an alternative to custom feature design, some authors advocate learning or optimizing features directly from the data [11]. Not surprisingly, this approach typically requires large model architectures, and much larger (annotated) data sets than had previously been used in MIR research. Due to the high cost of acquiring annotated musical data, it has so far been difficult to apply these techniques in most MIR tasks. While some authors have advocated leveraging unlabeled data to “pre-train” feature representations [6], recent studies have shown that comparable or better performance can be achieved with random initialization and fully supervised training [9, 22]. Our goal in this work is to provide data augmentation tools which may ease the burden of sample complexity, and make data-driven methodology more accessible to the MIR community.

Specific instances of data augmentation can be found throughout the MIR literature, though they are not often identified as such, nor are they treated systematically in a unified framework. For example, it is common to apply circular rotations to chroma features to achieve key invariance when modeling chord quality [15]. Alternately, synthetic mixtures of monophonic instruments have been used to generate more difficult examples when training polyphonic transcription engines [13]. Some authors even leave the audio content unchanged and only modify labels during training, as exemplified by the *target smearing* method of Ullrich *et al.* for training structural boundary detectors [21].

Finally, recent studies have used degraded signals to evaluate the stability of existing methods for MIR tasks. The Audio Degradation Toolbox (ADT) was developed for this purpose, and was used to measure the impact of naturalistic deformations of audio on several tasks, including beat tracking, score alignment, and chord recognition [16].

Similarly, Sturm and Collins proposed the “Kiki-Bouba Challenge” as a way to determine whether statistical models of musical concepts actually capture the defining characteristics of the concept (*e.g.*, genre), or are over-fitting to spurious correlations [20].

In both of the studies cited above, models are fit to unmodified data, and evaluated in degraded conditions under the control of the experimenter. Data augmentation provides the converse of this setting: models are fit to degraded data, and evaluated on unmodified examples. The distinction between the two approaches is critical. The former attempts to measure the robustness of a system under synthetic conditions, while the latter attempts to improve robustness by *training* under synthetic conditions. Note that with data augmentation, the evaluation set is left untouched by the experimenter, so the resulting comparisons are unbiased with respect to the underlying distribution from which the data are sampled. While this does not directly measure robustness, it has been observed that data augmentation can improve generalization [10, 14].

3. DATA AUGMENTATION ARCHITECTURE

Our implementation takes substantial inspiration from the Audio Degradation Toolbox [16]. In principle, the ADT can be used directly for some forms of data augmentation simply by applying it to the training set rather than test set. However, we opted for an independent, Python-based implementation for a variety of reasons.

First, Python enables object-oriented design, allowing for structured, extensible, and reusable code. This in turn facilitates a simple interface shared across all *deformation objects*, and makes it easy for practitioners to combine or extend existing deformations.

Second, we use JAMS [12] both to transport and store track annotations, and as an internal data structure for processing. JAMS provides a unified interface to different annotation types, and a convenient framework to manage all annotations for a particular track. This simplifies the tasks of maintaining synchronization between audio and annotations, and implementing task-dependent annotation deformations. We also adapt JAMS sandbox fields to provide data provenance and facilitate reproducibility.

Finally, we borrow familiar software design patterns from the scikit-learn package [4], such as *transformers*, *pipelines*, and model serialization. These building blocks allow practitioners to quickly and easily assemble complex pipelines from small, conceptually simple components.

In the remainder of this section, we will describe the software architecture in more detail. Without loss of generality, we assume that an annotation (*e.g.*, instrument activations) is encoded as a collection of tuples: (*time*, *duration*, *value*, *confidence*). Note that instantaneous events can be represented with zero duration, while track-level annotations have full-track duration. The *value* field depends on the annotation type, and may encode strings, numeric quantities, or fully structured objects.

3.1 Deformation objects

At the core of our implementation is the concept of a *deformation object*. We will first describe deformation objects in terms of their methods and abstract properties. Section 3.1.1 follows with a concrete, but high-level example.

A deformation object implements one or more *transformation methods*, each of which applies to either audio, meta-data, or annotations. Parameters of the deformation are shared through a *state* object S . For example, S might contain the speed-up factor of a time-stretch, or the number of semi-tones in a pitch-shift. Each transformation method takes as input a pair (S, x) and returns the transformed audio, meta-data, or annotation x' . Decoupling the deformation object's instantiation from its state allows multiple tracks to be processed in parallel by the same object. Moreover, as described in Section 3.3, state objects are reusable, which promotes reproducibility.

Data augmentation often requires sampling or sweeping a set of deformation parameters, and instantiating a separate deformation object for each parameterization can be inefficient, especially when the S contains non-trivial data (e.g., tuning estimates or noise signals). Instead, a deformation object implements a *state generator*, which may execute arbitrary transition logic to produce a sequence of states (S_1, S_2, \dots) . This is implemented efficiently using Python *generators*.

Finally, deformation objects may register transformation functions against the *type* of an annotation, as described by regular expressions. This allows different transformation procedures to be applied to different annotation types. During execution, the JAMS object is queried for all annotations matching the specified expression, and the results are processed by the corresponding transformation method. For example, the expression “.” matches all annotation types, while “chord.” matches only chord-type annotations. These patterns need not be unique or disjoint, though care must be taken to ensure consistent behavior. Deformations are always applied following the order in which they are registered.

The abstract transformation algorithm is described in Algorithm 1. For each state S , the input data J is copied, transformed into J' , and yielded back to the caller. Each J' can then be exported to disk, provided as a sample to an iterative learning algorithm, or passed along to another deformation object in a pipeline for further processing. When all subsequent processing of J' has completed, Algorithm 1 may resume computation at line 10 and proceed to the next state at line 2. Note that because deformation objects are both iterative (per track) and can be parallelized (across tracks), batches of deformed data can be generated online for stochastic learning algorithms.

3.1.1 Example: randomized time-stretching

To illustrate the deformation object interface, we will describe the implementation of a randomized *time-stretch* deformation object. In this case, each *state* object contains a single quantity: the stretch factor r . Algorithm 2 illustrates the state-generation logic for a randomized time-stretcher,

Algorithm 1 Abstract transformation pseudocode

```

Input: Deformation object  $D$ , JAMS object  $J$ 
Output: Sequence of transformed JAMS objects  $J'$ 
1: function  $D$ .TRANSFORM( $J$ )
2:   for states  $S \in D$ .STATES( $J$ ) do
3:      $J' \leftarrow \text{COPY}(J)$ 
4:      $J'$ .audio  $\leftarrow D$ .AUDIO( $S, J$ .audio)
5:      $J'$ .meta  $\leftarrow D$ .METADATA( $S, J$ .meta)
6:     for transformations  $g$  in  $D$  do
7:       for annotations  $A \in J'$  which match  $g$  do
8:          $J'.A \leftarrow g(S, A)$ 
9:      $J'$ .history  $\leftarrow \text{APPEND}(J'.\text{history}, S)$ 
10:    yield  $J'$ 

```

Algorithm 2 Randomized time-stretch state generator

```

Input: JAMS object  $J$ , number of deformations  $n$ , range bounds  $(r_-, r_+)$ 
Output: Sequence of states  $S$ 
1: function RANDOMSTRETCH.STATES( $J, \{n, r_-, r_+\}$ )
2:   for  $i$  in  $1, 2, \dots, n$  do
3:     Sample  $r \sim U[r_-, r_+]$ 
4:     yield  $S = \{r\}$ 

```

in which some n examples are generated by sampling r uniformly at random from an interval $[r_-, r_+]$.²

The JAMS object J over which the deformations will be applied is also provided as input to the state generator. Though not used in this example, access to J allows the state generator to pre-compute quantities of interest, such as track duration — necessary to ensure well-defined outputs from target-smearing deformations — or tuning estimates, which are used by pitch-shift deformations to determine when a shift is large enough to alter note labels.

Once a state S has been generated, the AUDIO() deformation method — D .AUDIO(S, J .audio) — applies the time-stretch to the audio signal, which is stored within the JAMS sandbox upon instantiation.³ Similarly, track-level meta-data can be modified by the METADATA() method. In this example, time-stretching will change the track duration, which is recorded in the JAMS meta-data field.

Next, a generic *annotation* deformation would be registered to the pattern “.” and apply the stretch factor to all *time* and *duration* fields of all annotations. This deformation would leave the annotation *values* untouched, since not all annotation types have time-dependent values.

Finally, any annotations whose *value* fields depend on time, such as *tempo*, can be modified directly by registering the transformation function against the appropriate type pattern, e.g., “tempo”. Other time-dependent type deformations would be registered separately as needed.

The time-stretching example is simple, but it serves to illustrate the flexibility of the architecture. It is straightforward to extend this example into more sophisticated de-

² The parameters n, r_-, r_+ are actually properties of the deformation object, but are listed here as method parameters to simplify exposition.

³ The *sandbox* provides unstructured storage space within a JAMS object, which is used in our framework as a scratch space for audio signals.

formations with structured state generators to sweep over deterministic parameter grids. For example, an additive background noise deformation could be parameterized by a collection of noise sources and a range of gain parameters, and generate one example for each unique combination of source and gain.

3.2 Pipelines and bypasses

Algorithm 1 describes the process by which a deformation object turns a single annotated audio example into a sequence of deformed examples. If we were interested in experimenting with only a single type of augmentation (e.g., time stretching), this would suffice. However, some applications may require combining or cascading multiple types of deformation, and we prefer a unified interface that obviates the need for customized data augmentation scripts.

Here, we draw inspiration from scikit-learn in defining *pipeline* objects. The general idea is simple: two or more deformation objects D_i can be chained together, and treated as a single, integrated deformation object. More precisely, for a deformation pipeline P composed of k stages:

$$P = (D_1, D_2, \dots, D_k),$$

examples are generated by a depth-first traversal of the Cartesian product of the corresponding state spaces Σ_i :

$$\Sigma_P = \Sigma_1 \times \Sigma_2 \times \dots \times \Sigma_k.$$

One input example therefore produces $|\Sigma_P| = \prod_{i=1}^k |\Sigma_i|$ outputs. By using generators rather than explicit lists of states, we ensure that only $k + 1$ examples (counting the input) are ever in memory at any time. In most cases, k is much smaller than $|\Sigma_P|$, which provides substantial improvements to memory efficiency.

Finally, we introduce the *bypass* object, which is used to mark individual pipeline stages as optional. Bypasses are useful when it is difficult to encode a special *no transformation* state within a deformation object, such as in the randomized time-stretch example of Algorithm 2. The internal logic of a bypass object is simple: first, pass the input directly through unmodified, and then generate samples from the contained deformation object as usual. Bypasses can be used to ensure that the original examples are propagated through the pipeline unscathed, and the resulting augmented data set is a strict superset of the clean data.

3.3 Reproducibility and data provenance

When modifying data for statistical modeling purposes, maintaining transparency is of utmost importance to ensure reproducibility and accurate interpretation of results. This ultimately becomes a question of data provenance [5]: a record of all transformations should be kept, preferably attached as closely as possible to the data. Rather than force practitioners to handle book-keeping, we automate the process from within the deformation engine. This is accomplished at line 9 of Algorithm 1 by embedding the *state* object S (and, in practice, the parameters used to construct the deformation object D) within the JAMS object

Table 1. The 15 instrument labels used in our experiments.

Instrument	# Tracks	# Artists
drum set	65	57
electric bass	64	53
piano	42	23
male singer	38	34
clean electric guitar	37	32
vocalists	27	25
synthesizer	27	21
female singer	25	17
acoustic guitar	24	16
distorted electric guitar	21	20
auxiliary percussion	18	17
double bass	16	13
violin	14	5
cello	11	8
flute	11	6

after each deformation is applied. Each J' generated at line 10 thus contains a full transactional history of all modifications required to transform J into J' . For this reason, stochastic deformations are designed so that all randomness is contained within the state generator, and transformations are all deterministic.

In addition to facilitating reproducibility, maintaining transformation provenance allows practitioners to compute a wide range of deformations, and later filter the results to derive subsets generated by different augmentation parameters.

To further facilitate reproducibility and sharing of experimental designs, the proposed architecture supports *serialization* of deformation objects and pipelines into a simple, human-readable JavaScript object notation (JSON) format. Once a pipeline has been constructed, it can be exported, edited as plain text, shared, and reconstructed. This feature also simplifies the process of applying several different sets of deformation parameters, and eliminates the need for writing a custom script for each setting.

4. EXAMPLE: INSTRUMENT RECOGNITION

We applied data augmentation to the task of instrument recognition in mixed audio signals. For this task, we used the MedleyDB dataset, which consists of 122 tracks, spanning a variety of genres and instrumentation [3]. Each track is strongly annotated with time-varying instrument activations derived from the recording stems. MedleyDB is a small, but well-annotated collection, which we selected because it should be possible to over-fit with a reasonably complex model. Our purpose here is not to achieve the best possible recognition results, but to investigate utility of data augmentation for improving generalization. However, because of the small sample size, we limited the experiment to cover only the 15 instruments listed in Table 1.

For evaluation purposes, each test track is split into disjoint one-second clips. The system is then tasked with recognizing the instruments active within each clip. The system is evaluated according to the average track-wise mean (label-ranking) average precision (LRAP), and per-

instrument F -score over one-second clips.

4.1 Data augmentation

The data augmentation pipeline consists of four stages:

Pitch shift by $n \in \{-1, 0, +1\}$ semitones.

Time stretch by a factor of $r \in \{2^{-1/2}, 1.0, 2^{1/2}\}$.

Background noise (bypass) under three conditions: subway, crowded concert hall, and night-time city noise. Noise clips were randomly sampled and linearly mixed with the input signal y using random weights

$$\alpha \sim U[0.1, 0.4];$$

$$y' \leftarrow (1 - \alpha) \cdot y + \alpha \cdot y_{\text{noise}}.$$

Dynamic range compression (bypass) under two settings drawn from the Dolby E standards [7]: *speech*, and *music (standard)*.

Pitch-shift and time-stretch operations were implemented with Rubberband [1], and dynamic range compression was implemented using the *comand* function of sox [2]. Note that the first two stages include null parameter settings $n = 0$ and $r = 1$. Bypasses on the final two stages ensure that all combinations of augmentation are present in the final set. The full pipeline produces

$$|\Sigma_P| = 3 \times 3 \times (3 + 1) \times (2 + 1) = 108$$

variants of each input track. To simplify the experiments, we only compare the cumulative effects of the above augmentations. This results in five training conditions of increasing complexity:

- (N) no augmentation,
- (P) pitch shift,
- (PT) pitch shift and time stretch,
- (PTB) pitch shift, time stretch, and noise,
- (PTBC) all stages.

4.2 Acoustic model

The acoustic model used in these experiments is a deep convolutional network. The input to the network consists of log-amplitude, constant-Q spectrogram patches extracted with librosa [17]. Each example spans approximately one second of audio, corresponding to 44 frames at a hop length of 512 samples and sampling rate of 22050 Hz. Constant-Q spectrograms cover the range of C2 (65.41 Hz) to C8 (4186 Hz) at 36 bins per octave, resulting in time-frequency patches $X \in \mathbb{R}^{216 \times 44}$. Instrument activations are aggregated into a single binary label vector, such that an instrument is deemed active if its on-time within the sample exceeds 0.25 seconds.

Constant-Q representations are linear in both time and pitch, a property that can be exploited by convolutional neural networks to achieve translation invariance. Thus a four-layer model is designed to estimate the presence of

zero or more instruments in a time-frequency patch. Formally, an input X , is transformed into an output Z , via a composite nonlinear function $\mathcal{F}(\cdot; \Theta)$ with parameters Θ . This is achieved as a sequential cascade of $L = 4$ operations, $f_\ell(\cdot; \theta_\ell)$, referred to as *layers*, the order of which is given by ℓ :

$$Z = \mathcal{F}(X | \Theta) = f_L(\dots f_2(f_1(X | \theta_1) | \theta_2) | \theta_L) \quad (1)$$

The first two layers, $\ell \in \{1, 2\}$, are convolutional, expressed by the following:

$$Z_\ell = f_\ell(X_\ell | \theta_\ell) = h(W \otimes X_\ell + b), \quad \theta_\ell = [W; b] \quad (2)$$

Here, the valid convolution, \otimes , is computed by convolving a 3D input tensor, X_ℓ , consisting of N feature maps, with a collection of M 3D-kernels, W , followed by an additive vector bias term, b , and transformed by a point-wise activation function, $h(\cdot)$. In this formulation, X_ℓ has shape (N, d_0, d_1) , W has shape (M, N, m_0, m_1) , and the output, Z_ℓ , has shape $(M, d_0 - m_0 + 1, d_1 - m_1 + 1)$. Max-pooling is applied in time and frequency, to further accelerate computation by reducing the size of feature maps, and allowing a small degree of scale invariance in both time and pitch.

The final two layers, $\ell \in \{3, 4\}$, are fully-connected matrix products, given as follows:

$$Z_\ell = f_\ell(X_\ell | \theta_\ell) = h(WX_\ell + b), \quad \theta_\ell = [W; b] \quad (3)$$

The input to the ℓ^{th} layer, X_ℓ , is flattened to a column vector of length N , projected against a weight matrix W of shape (M, N) , added to a vector bias term, b , of length M , and transformed by a point-wise activation function, $h(\cdot)$.

The network is parameterized thusly: ℓ_1 uses W with shape $(24, 1, 13, 9)$, followed by $(2, 2)$ max-pooling over the last two dimensions, and a rectified linear unit (ReLU) activation function: $h(x) := \max(x, 0)$; ℓ_2 has filter parameters W with shape $(48, 24, 9, 7)$, followed by $(2, 2)$ max-pooling over the last two dimensions, and a ReLU activation function; ℓ_3 uses W with shape $(17280, 96)$ and a ReLU activation function; finally, ℓ_4 uses W with shape $(96, 15)$ and a sigmoid activation function.

During training, the model optimizes cross-entropy loss via mini-batch stochastic gradient descent, using batches of $n = 64$ randomly selected patches and a constant learning rate of 0.01. Dropout is applied to the activations of the penultimate layer, $\ell = 3$ with dropout probability 0.5. Quadratic regularization is applied to the weights of the final layer, $\ell = 4$, with a penalty factor of 0.02. This helps prevent numerical instability by keeping the weights from growing arbitrarily large. The model is check-pointed after every 1000 batches (up to 50000 batches), and a validation set is used to select the parameter setting achieving the highest mean LRAP.

4.3 Evaluation

Fifteen random artist-conditional partitions of the MedleyDB collection were generated with a train/test artist ratio of 4:1. For the purposes of this experiment, *MusicDelta* tracks were separated by genre into a collection of distinct

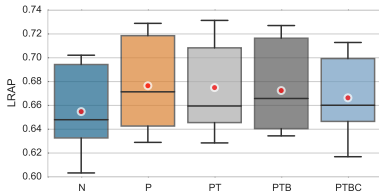


Figure 1. Test-set score distributions (mean track-wise label-ranking average precision), over all train-test splits. Mean scores are indicated by \bullet . Boxes cover the 25–75 percentiles, and whiskers cover the 5–95 percentiles.

pseudo-artists. This results in 75 unique artist identifiers for the 122 tracks. For each train/test split, the training set was further partitioned into training and validation sets, again at a ratio of 4:1. To evaluate performance, we compute for each test track the mean label-ranking average precision (LRAP) over all disjoint one-second patches.

4.3.1 Label ranking results

Figure 1 illustrates the distribution of test-set performance across splits. Between the no-augmentation condition (N) and pitch-shifting augmentation (P), there is a small, but consistent improvement in performance from an average of 0.655 to 0.677. This is in keeping with the motivation for this work, and our expectations when training a (pitch)-convolutional model on a small sample. If the amount of clean data is too small, the model may easily over-fit by capturing irrelevant, correlated properties. (For example, if all of the *piano* recordings are in one key, the model may simply capture the key rather than the characteristics of *piano*.) Adding pitch-shifted examples should help the model disambiguate these properties.

Subsequent deformations do not appear to improve over condition (P). In each case, no significant difference from the pitch-shift condition could be detected by a Bonferroni-corrected Wilcoxon signed-rank test. However, all deformation conditions consistently outperform the baseline (N).

Although the difference in average performance is relatively small, the upper and lower quantiles are notably higher in (P), (PT), and (PTB) conditions. This indicates a reduction in the tendency to over-fit the relatively small training sets used in these experiments.

4.3.2 Frame-tagging results

To investigate the effects of augmentation on each instrument class, we computed the F -score of frame-level instrument recognition under each training condition. Results were averaged first across test tracks in a split, and then across all splits. Figure 2 depicts the change in F -score relative to the baseline condition (N): $\Delta F = (F - F_N)$.

The trend is primarily positive: in all but three classes, all augmentation conditions provide consistent improvement. The three exceptions are *synthesizer*, *female singer*, and *violin*. In the latter two cases, negative change is only observed after introducing time-stretch deformations, which

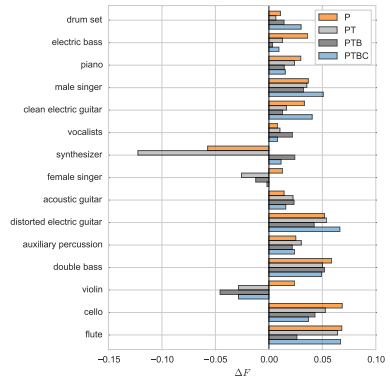


Figure 2. Per-class change in mean test-set F -score for each augmentation condition (F), relative to the no-augmentation baseline (F_N).

may unnaturally distort vibrato characteristics and render these classes more difficult to model. The effect is particularly prominent for *violin*, which has the fewest unique artists, and produces the fewest training examples.

The reduction in F -score for *synthesizer* in the (PT) condition may explain the corresponding reduction in Figure 1, and may be due to a confluence of factors. First, many of the synthesizer examples in MedleyDB have low amplitudes in the mix, and may be difficult to model in general. Second, the class itself may be ill-defined, as *synthesizer* encompasses a range of instruments and timbres which may be artist-dependent and idiosyncratic. Simple augmentations can have adverse effects if the perturbed examples are insufficiently varied from the originals, which may be the case here for (P) and (PT). However, the inclusion of background noise (PTB) results in a slight improvement over the baseline.

5. CONCLUSION

The data augmentation framework provides a simple and flexible interface to train models on distorted data. The instrument recognition experiment demonstrates that even simple deformations such as pitch-shifting can improve generalization, but that some care should be exercised when selecting deformations depending on the characteristics of the problem. We note that these results are preliminary, and do not fully exploit the capabilities of the augmentation framework. In future work, we will investigate the data augmentation for a variety of MIR tasks.

6. ACKNOWLEDGEMENTS

BM acknowledges support from the Moore-Sloan Data Science Environment at NYU. This material is partially based upon work supported by the National Science Foundation, under grant IIS-0844654.

7. REFERENCES

- [1] Rubber band library v1.8.1, October 2012. <http://rubberbandaudio.com/>.
- [2] sox v14.4.1, February 2013. <http://sox.sourceforge.net/>.
- [3] Rachel Bittner, Justin Salamon, Mike Tierney, Matthias Mauch, Chris Cannam, and Bello, Juan Pablo. MedleyDB: a multitrack dataset for annotation-intensive mir research. In *15th International Society for Music Information Retrieval Conference*, ISMIR, 2014.
- [4] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, et al. API design for machine learning software: experiences from the scikit-learn project. In *European Conference on Machine Learning and Principles and Practices of Knowledge Discovery in Databases*, 2013.
- [5] Peter Buneman, Sanjeev Khanna, and Wang-Chiew Tan. Data provenance: Some basic issues. In *FST TCS 2000: Foundations of software technology and theoretical computer science*, pages 87–93. Springer, 2000.
- [6] Sander Dieleman, Philémon Brakel, and Benjamin Schrauwen. Audio-based music classification with a pretrained convolutional network. In *12th international society for music information retrieval conference*, ISMIR, 2011.
- [7] Dolby Laboratories, Inc. *Standards and practices for authoring Dolby Digital and Dolby E bitstreams*, 2002.
- [8] Daniel PW Ellis and Thierry Bertin-Mahieux. Large-scale cover song recognition using the 2d fourier transform magnitude. In *The 13th international society for music information retrieval conference*, ISMIR, 2012.
- [9] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier networks. In *Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. JMLR W&CP Volume*, volume 15, pages 315–323, 2011.
- [10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *arXiv preprint arXiv:1502.01852*, 2015.
- [11] Eric J Humphrey, Juan Pablo Bello, and Yann LeCun. Moving beyond feature design: Deep architectures and automatic feature learning in music informatics. In *The 13th international society for music information retrieval conference*, ISMIR, 2012.
- [12] Eric J Humphrey, Justin Salamon, Oriol Nieto, Jon Forsyth, Rachel M Bittner, and Bello, Juan Pablo. JAMS: A JSON annotated music specification for reproducible MIR research. In *15th International Society for Music Information Retrieval Conference*, ISMIR, 2014.
- [13] Holger Kirchhoff, Simon Dixon, and Anssi Klapuri. Multi-template shift-variant non-negative matrix deconvolution for semi-automatic music transcription. In *The 13th international society for music information retrieval conference*, ISMIR, 2012.
- [14] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, NIPS, pages 1097–1105, 2012.
- [15] Kyogu Lee and Malcolm Slaney. Acoustic chord transcription and key extraction from audio using key-dependent HMMs trained on synthesized audio. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):291–301, 2008.
- [16] Matthias Mauch and Sebastian Ewert. The audio degradation toolbox and its application to robustness evaluation. In *14th International Society for Music Information Retrieval Conference*, ISMIR, 2013.
- [17] Brian McFee, Matt McVicar, Colin Raffel, Dawen Liang, Dan Ellis, Douglas Repetto, Petr Viktorin, and Joo Felipe Santos. *librosa: 0.4.0rc1*, March 2015.
- [18] Meinard Müller and Sebastian Ewert. Chroma toolbox: Matlab implementations for extracting variants of chroma-based audio features. In *12th International Conference on Music Information Retrieval*, ISMIR, 2011.
- [19] Elias Pampalk. A matlab toolbox to compute music similarity from audio. In *International Symposium on Music Information Retrieval (ISMIR2004)*, 2004.
- [20] Bob L Sturm and Nick Collins. The Kiki-Bouba Challenge: Algorithmic composition for content-based MIR Research & Development. In *International Symposium on Music Information Retrieval*, 2014.
- [21] Karen Ullrich, Jan Schlüter, and Thomas Grill. Boundary detection in music structure analysis using convolutional neural networks. In *15th International Society for Music Information Retrieval Conference*, ISMIR, 2014.
- [22] Matthew D Zeiler, M Ranzato, Rajat Monga, M Mao, K Yang, Quoc Viet Le, Patrick Nguyen, A Senior, Vincent Vanhoucke, Jeffrey Dean, et al. On rectified linear units for speech processing. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3517–3521. IEEE, 2013.

Oral Session 2
Rhythm & Beat

DRUM TRANSCRIPTION USING PARTIALLY FIXED NON-NEGATIVE MATRIX FACTORIZATION WITH TEMPLATE ADAPTATION

Chih-Wei Wu, Alexander Lerch

Georgia Institute of Technology, Center for Music Technology
{cwu307, alexander.lerch}@gatech.edu

ABSTRACT

In this paper, a template adaptive drum transcription algorithm using partially fixed Non-negative Matrix Factorization (NMF) is presented. The proposed method detects percussive events in complex mixtures of music with a minimal training set. The algorithm decomposes the music signal into two dictionaries: a percussive dictionary initialized with pre-defined drum templates and a harmonic dictionary initialized with undefined entries. The harmonic dictionary is adapted to the non-percussive music content in a standard NMF procedure. The percussive dictionary is adapted to each individual signal in an iterative scheme: it is fixed during the decomposition process, and is updated based on the result of the previous convergence. Two template adaptation methods are proposed to provide more flexibility and robustness in the case of unknown data. The performance of the proposed system has been evaluated and compared to state of the art systems. The results show that template adaptation improves the transcription performance, and the detection accuracy is in the same range as more complex systems.

1. INTRODUCTION

Being one of the most intensively researched areas in Music Information Retrieval (MIR), automatic music transcription is often considered the core technology that would enable high-level representations of music signals with the potential of improving virtually any MIR system. A complete transcription system comprises many sub-tasks such as multi-pitch detection, onset detection, instrument recognition, and rhythm extraction [2]. While the main focus is mostly on pitched instruments, a considerable amount of publications deal with the transcription of percussive sounds in mixtures of tonal and percussive instruments. The drum track in popular music conveys information about tempo, rhythm, style, and possibly the structure of a song. A drum transcription system enables applications in active listening [27], music education, and interactive music performance.

This study explores the application of the popular transcription method NMF for drum transcription in polyphonic music. A standard NMF approach for music transcription decomposes a signal into a dictionary matrix, which consists of multiple pre-defined templates, and an activation matrix, which contains the activity of the corresponding templates. In this paper, we propose to transcribe drum events using a signal-adaptive method based on NMF.

The paper is structured as follows: Section 2 provides an overview of the research in this area. In Section 3 we present our approach; evaluation results are being presented and discussed in Section 4. Section 5 provides a summary, conclusion, and directions of future work.

2. RELATED WORK

Drum transcription is a task that requires instrument identification and onset detection for percussive sounds. To transcribe signals containing only drum sounds, standard approaches with a feature extractor and a subsequent classifier are able to produce results with high accuracy [11]. For most use cases, however, a drum transcription system is expected to work on mixtures of percussive and harmonic sound sources. Gillet and Richard propose to categorize automatic drum transcription systems into three categories: (i) *segment and classify* [4, 7, 22], for which the audio signal is segmented into a series of events using onset detection, and each event is classified based on the extracted temporal or spectral features, (ii) *separate and detect* [1, 6, 15, 17], which assumes music to be a superposition of different sound sources; by decomposing the signal into source templates with corresponding activation functions, the content can be transcribed by analyzing the activities of each template, and (iii) *match and adapt* [28, 29], identifying the drum events using a template matching method in which the templates are searched for the closest match and adapted in an iterative process.

Methods extended from these three types of approaches have been presented as well. Paulus and Klapuri proposed to use Hidden Markov Models (HMM) for drum transcription [16]. This method models temporal connections between drum events and detect the drum based on the probabilistic model. However, the method needs to train on multiple drum sequences, thus, a large dataset is needed to obtain a generic model. Another recent approach is to use bar information to classify the audio signal into different predefined drum patterns [23]. This approach requires addi-



© Chih-Wei Wu, Alexander Lerch.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Chih-Wei Wu, Alexander Lerch. "Drum Transcription Using Partially Fixed Non-Negative Matrix Factorization with Template Adaptation", 16th International Society for Music Information Retrieval Conference, 2015.

tional information of the bar locations and a large dictionary, which can be impractical in some use cases.

Among the above mentioned methods, the second type of approaches (*separate and detect*), frequently using NMF-related methods, has the advantage of joint estimation of multiple instruments and easy interpretation of the results. However, when NMF is applied to the task of drum transcription, the following challenges have to be faced:

First, the number of sound sources and notes within a music recording is usually unknown. To optimally decompose a signal, this number is necessary for determining the rank r of the dictionary. This problem would be less severe when the sound sources of the target signal are given [14]. However, in most cases, this prior information is difficult to acquire. One solution is to build a dictionary that contains more source templates than the target signal. Benetos et al. used a probabilistic extension of NMF (Probabilistic Latent Component Analysis, PLCA) to jointly transcribe pitched and unpitched sounds in polyphonic music with a relatively large pre-trained dictionary [3]. Although this method can provide harmonic and percussive contents of the music simultaneously, its robustness against unknown sources still needs to be evaluated.

Second, without any prior knowledge, it can be hard to identify the corresponding instrument of every template in the dictionary matrix [26]. This problem becomes more severe when the rank is selected too high or too low. Helen and Virtanen trained an SVM to separate drum templates from harmonic templates; the rank number was derived empirically during the factorization process [10]. The identified drum templates and their corresponding activation could later be used to reconstruct the drum signal, resulting in a system for drum source separation. Their approach requires a significant amount of training data for the classifier and, more importantly, the results can be expected to be very susceptible to choice of rank. Yoo et al. proposed a co-factorization algorithm [26] to simultaneously factorize a drum track and a polyphonic signal. They used the dictionary matrix from the drum track to identify the drum templates in the polyphonic signal. This approach ensures that the drum templates in both dictionary matrices are estimated only from the drum track, resulting in proper isolation of the harmonic templates from the drum templates. Since their system aims at drum separation, they can work at higher ranks. For drum transcription, however, this approach is not directly applicable because the corresponding instrument of the templates in the dictionary matrix is unknown.

Third, a suitable penalty term or sparsity constraint for detecting percussive instruments still needs to be investigated. In general, these constraints are the additional terms in the NMF cost function that will facilitate the different properties (e.g., the sparseness) in the resulting activation matrix. Virtanen proposed to use constraints for temporal continuity and sparseness [24]. He reported that by using the temporal continuity criterion, the detection accuracy and SNR of the pitched sounds can be improved in the source separation task, whereas no significant improvement

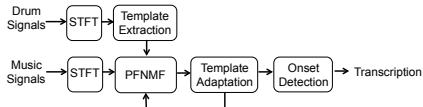


Figure 1. Flowchart of the drum transcription system

is shown with the sparseness constraint.

Another issue is the adaptability of the extracted templates. When using supervised NMF, the algorithm loses its adaptability and might fail when the target signal is very different from the pre-trained dictionary. Dittmar and Gartner proposed to use semi-adaptive bases during the NMF decomposition process [5]. However, their results indicate that the semi-adaptive process did not improve the performance of the transcription accuracy compared to fixed bases. Furthermore, no results were reported for the transcription performance in polyphonic mixtures.

3. METHOD

3.1 Implementation

Figure 1 shows the flow chart of the implemented system. The STFT of the signals will be calculated using a Hann window with a window length and a hop size of 2048 and 512, respectively, and the sample rate is 44.1 kHz. The resulting magnitude spectrogram is used as the input representation. A pre-trained dictionary matrix W_D will be constructed from the training set, which consists of isolated drum sounds. Next, the initial drum dictionary will be used in the partially fixed NMF (PFNMF) process and updated by the selected template adaptation methods described in Section 3.3. Finally, the activation matrix H_D is processed to determine the onset positions and their corresponding classes.

The initial drum dictionary matrix W_D is generated from a subset of the ENST dataset, which contains audio tracks of 5 to 6 single hits for each drum, performed by three drummers. For every drum class, one track per drummer is collected as training data. The onset position of these single hits was determined using the annotated ground truth. The template spectrum is a median spectrum of all individual events of one drum class in the training set. The templates are extracted for the three classes: Hi-Hat (HH), Bass Drum (BD) and Snare Drum (SD).

High values in the activation matrix H_D indicate the presence of a drum event. More specifically, the activity difference of each row of the activation matrix could be considered as the onset novelty function of each individual drum. We use a median filter as a standard approach to create a signal-adaptive threshold for peak picking [13]. In this paper, the window length and the offset coefficient λ of the median adaptive threshold are set to be 0.1 s and 0.12 for every track. The Matlab implementation of the presented system is available online.¹

¹ <https://github.com/cwu307/NmfDrumToolbox>

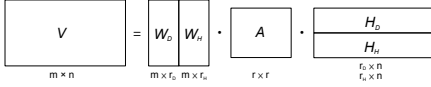


Figure 2. Illustration of the factorization process. W : dictionary matrix, H : activation matrix; Subscript D : drum, subscript H : harmonic components. A is the weighting matrix.

3.2 Algorithm Description

The basic concept of NMF can be expressed as $V \approx WH$ with non-negativity constraints, in which V is a $m \times n$ matrix, W is a $m \times r$ dictionary matrix, and H is a $r \times n$ activation matrix, with r being the rank of the NMF decomposition. In most audio applications, V is the spectrogram with m frequency bins and n frames. W contains the magnitude spectra of the salient components, and H indicates the activation of these components with respect to time [20]. The matrices W and H are estimated through an iterative process that minimizes a distance measure between the target spectrogram V and its approximation [12].

In this paper, we propose a signal adaptive method to transcribe drum events in polyphonic signals. The idea of using NMF with prior knowledge of the target source within the mixture has been applied to source separation tasks [21] and multipitch analysis [18]. The method described here is based on similar ideas but with different emphasis: (i) we focus on a real world scenario in which users only have limited amount of training samples that are slightly different from the target source, (ii) we propose to use a small dictionary matrix which is both efficient and easily interpretable, and (iii) the proposed method is able to adapt to different content in the polyphonic mixtures.

PFNMF [25] is a method inspired by [26] for drum transcription task. Figure 2 visualizes the concept: the matrices W and H are split into the matrices W_D and W_H , and H_D and H_H , respectively. Instead of using co-factorization, the algorithm initializes the matrix W_D with drum templates and does not modify it during the factorization process. The matrices W_H , H_H , and H_D are initialized randomly. The rank r_D of W_D and H_D depends on the number of templates (i.e., instruments) provided, and the rank r_H can be arbitrarily chosen. The total rank $r = r_D + r_H$. A is a $r \times r$ diagonal weighting matrix, which contains weighting coefficients for every template to balance the drum and harmonic dictionaries in the NMF cost function (as discussed in Section 4.3.1). In our experiment, the coefficients are set to be $\alpha = (r_D + r_H)/r_D$ for each drum template and $\beta = r_H/(r_D + r_H)$ for each harmonic template. This setting is to increase the weighting of drum templates and slightly decrease the weighting of harmonic templates as r_H becomes larger. When $r_H = 0$, the algorithm reduces to the original NMF.

The distance measure used is KL-divergence, in which $D_{KL}(x || y) = x \cdot \log(x/y) + (y - x)$. The NMF cost function as shown in Eq. (1) is minimized by applying

gradient decent and multiplicative update rules.

$$J = D_{KL}(V || \alpha W_D H_D + \beta W_H H_H) \quad (1)$$

The matrices W_H , H_H , and H_D will be updated according to Eqs. (2)–(4):

$$H_D \leftarrow H_D \frac{W_D^T (V / (\alpha W_D H_D + \beta W_H H_H))}{W_D^T} \quad (2)$$

$$W_H \leftarrow W_H \frac{(V / (\alpha W_D H_D + \beta W_H H_H)) H_H^T}{H_H^T} \quad (3)$$

$$H_H \leftarrow H_H \frac{W_H^T (V / (\alpha W_D H_D + \beta W_H H_H))}{W_H^T} \quad (4)$$

To summarize, the presented method before template adaptation consists of the following steps:

1. Construct a $m \times r_D$ dictionary matrix W_D , with r_D being the number of drum components to be detected.
2. Given a pre-defined rank r_H , initialize a $m \times r_H$ matrix W_H , a $r_D \times n$ matrix H_D and a $r_H \times n$ matrix H_H .
3. Normalize W_D and W_H .
4. Update H_D , W_H , and H_H using Eqs. (2)–(4).
5. Calculate the cost of the current iteration using Eq. (1).
6. Repeat step 3 to step 5 until convergence.

The time positions of the drum events can then be extracted by applying a simple onset detection on the rows of matrix H_D .

3.3 Template Adaptation

Previous approaches to include template adaptation in drum transcription process can be found in [5, 29]. These approaches usually start with seed templates and gradually adapt them to the optimal templates. In this paper, we propose two methods for template adaptation with PFNMF. Both methods have the same criterion to stop iterating when the error between two consecutive iterations changes by less than 0.1% or the number of iterations exceeds 20. However, the adaptation process typically converges after 5–10 iterations.

3.3.1 Method 1: Complementary Update

In the first method (referred to as AM1), the drum dictionary W_D is updated based on the cross-correlation between the activations H_H and of each individual drum in H_D . PFNMF starts by randomly initializing a W_H with rank r_H . Although W_H tends to adapt to the harmonic content, it may still contain entries that belong to percussive instruments due to a mismatch between the initialized drum templates and the target sources. This will result in cross-talk (simultaneous activation) between H_H and H_D and generate a less pronounced activation. However, these harmonic templates may also provide complementary information to the original drum templates. To identify these entries, the normalized cross-correlation between H_H and H_D for each individual drum is computed using Eq. (5)

$$\rho_{x,y} = \frac{\sum_{j=1}^n x(j) \cdot y(j)}{\|x\|_2 \cdot \|y\|_2}, \quad (5)$$

where x and y represent different activation vectors, and n is the number of samples in the activation vectors. A threshold ρ_{thres} is defined for identification of related entries, and the drum template W_D can be updated using Eq. (6), where $W_H^{(i)}$ ($i = 1, \dots, S$) are the entries with their corresponding $\rho_{x,y}$ higher than ρ_{thres} , and S is the number of the selected entries. Since a low ρ_{thres} can introduce too much adaptation and vice versa, a $\rho_{thres} = 0.5$ is chosen heuristically. The amount of adaptation also depends on the coefficient $\gamma = \frac{1}{2^k}$, which decreases as iteration number k increases.

$$W'_D = (1 - \gamma)W_D + \gamma \frac{1}{S} \sum_{i=1}^S \left(\rho^{(i)} W_H^{(i)} \right) \quad (6)$$

3.3.2 Method 2: Alternate Update

In the second method (referred to as AM2), the drum template W_D is adapted by alternatively fixing W_D and H_D during the decomposition process. The adaptation process starts by fixing W_D , and PFNMF will try to fit the best activation H_D to approximate the drum part in the music. Once H_D is determined, a new iteration of PFNMF can be started by fixing H_D and allow W_D , W_H and H_H to update. This constraint will guide the algorithm to fit better drum templates based on the detected activation H_D . The update rule for W_D is shown in Eq. (7).

$$W_D \leftarrow W_D \frac{(V/(\alpha W_D H_D + \beta W_H H_H)) H_D^T}{H_D^T} \quad (7)$$

4. EVALUATION

4.1 Dataset Description

The experiments have been conducted on two different datasets. The first one is the *minus one* subset from the ENST drum dataset [8]. This dataset consists of recordings from three different drummers performing on their own drum kits. The set for each drummer contains individual hits, short phrases of drum beats, drum solos, and short excerpts played with accompaniments. The minus one subset has 64 tracks of polyphonic music, and the sampling rate of every track is 44.1 kHz. Each track in this subset has a length of approximately 70 s with varying style. More specifically, the subset contains various drum playing techniques such as ghost notes, flam, and drag; these techniques are considered difficult to identify with existing drum transcription systems [9]. The accompaniments are mixed with their corresponding drum tracks using a scaling factor of 1/3 and 2/3 in order to reproduce the evaluation settings as used in [16].

The second dataset, used for cross-dataset validation, is IDMT-SMT-Drums [5]. This dataset consists of 95 drum loop recordings from three drum kits (RealDrum, WaveDrum and TechnoDrum). The sampling rate of every track is 44.1 kHz, and the total duration of the dataset is approximately two hours. This dataset also contains isolated drum hits for training. However, in our experiments, the isolated sounds are not used.

4.2 Evaluation Procedure

We evaluate the proposed system for both monophonic (drum only) and polyphonic mixtures. The same set of audio tracks is used with and without accompaniments. A three-fold cross-validation is applied to the evaluation process. Single drum hits collected from two drummers are used to train the system, and complete mixtures from the third drummer are used to test the system. The process repeats three times to test every drummer in the dataset. This process is the same as described in [16], and the purpose is to prevent the system from seeing the test data. Note that the training data used in the system are single drum hits, and the number of onsets is significantly fewer than the test data. Typically, the training data only consists of 10 to 12 single hits for each drum class. This is similar to the real-world use case, where the users may have access only to a limited number of training samples.

The evaluation metrics follow the standard calculation of the precision (P), recall (R), and F-measure (F). To be consistent with [9], an onset is considered to be a match with the ground truth if the time deviation in between is less or equal to 50 ms. It should be noted that some authors use more restrictive settings, compare e.g. the 30 ms as used in [16].

4.3 Evaluation Results

4.3.1 Rank Independence

In an initial test to determine the rank r_H of the PFNMF, $r_H = 5, 10, 20, 40, 80, 160$ have been tested in polyphonic signals with and without a weighting matrix. As shown in Figure 3, a general trend of decreasing performance can be observed when $r_H > 5$ without a weighting matrix. With a weighting matrix, however, the performance slightly increases for both HH and SD, and slightly decreases for BD as the r_H increases. The results demonstrate the robustness of the proposed system against the rank selection when a weighting matrix is introduced.

By increasing the rank r_H , a larger W_H will be initialized to better adapt to the target signal, however, this unbalanced increase in templates would also decrease the weight of the drum templates in the optimization process, thus reducing the impact of the percussive templates on the NMF cost function. This effect is reduced by the weighting matrix A which balances the weights between drum and harmonic templates.

4.3.2 Threshold Selection

The transcription results can be obtained after applying onset detection on each drum activation (see Section 3.1). However, the performance varies according to the selection of the signal-adaptive threshold. To evaluate the influence of different thresholds, the average F-measure of all drums with different offset coefficient λ on IDMT-SMT-Drums dataset is shown in Figure 4. A general trend of parabolic curve can be observed. This is in agreement with the findings of Dittmar et al. [5]. One major difference is that in most regions of the curve, both AM1 and AM2 outperform

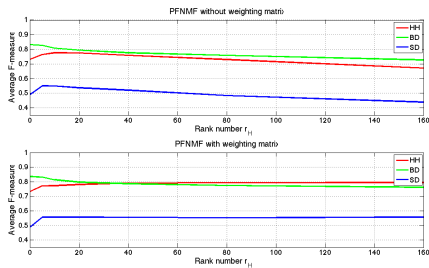


Figure 3. Average F-measure versus harmonic rank r_H in (Top) without weighting matrix (Bottom) with weighting matrix

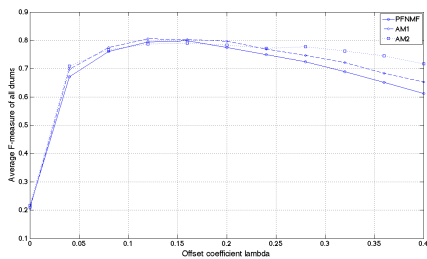


Figure 4. Evaluation results for IDMT-SMT-Drums dataset using (a) PFNMF (Solid circle) (b) AM1 (Dash diamond) (c) AM2 (Dotted square)

PFNMF. This verifies that template adaptation process does help the algorithm in the case of the unknown sounds (templates and the test signals are from two different datasets). The overall performance is slightly lower than [5] due to the mismatch in templates and target signals. However, the F-measures of AM1 can reach 74.0%, 93.2% and 73.4% for HH, BD, SD, respectively, which indicates the applicability of the proposed method across datasets.

4.3.3 Results

Table 1 shows the evaluation results on ENST drum dataset *minus one subset without accompaniments*. For comparison, we also list the results of Gillet et al. [9] and Paulus et al. [16]. All the compared methods use the same dataset with identical mixing settings (1/3 for accompaniments and 2/3 for drum tracks). Since the target signals contain only drum sounds, the rank r_H can be small. In this experiment, r_H is set to 10 for absorbing drum sounds other than HH, BD and SD. The results show that our proposed method is able to transcribe drum events with an average F-measure of 77.9% using AM2. This result is higher than the 73.8% reported in [9], and at the same level as reported in [16].

Table 2 shows the evaluation results on ENST drum dataset *minus one subset with accompaniments*. The compared methods are the same as described above. Since the target signals contain both percussive and harmonic parts, r_H is set to 50. The results show that our proposed method achieves an average F-measure = 72.2% using AM2, which is higher than 67.8% [9] and at a similar range as the 72.7%, reported in [16].

In general, our methods outperform [9] for all instruments except the snare drum. The possible reason is that many of the playing technique variations are applied to the snare (e.g., ghost note, rim shot, with/without snare on), and a single snare drum template cannot cover all the possibilities even with template adaptation. In the polyphonic dataset, our proposed methods perform better on BD and SD but slightly worse on HH compared to the HMM based method [16]. Since Paulus et al. [16] trained and tested their system using the same ENST dataset, the music played by all three drummers is highly correlated because of the same accompaniments used. This may lead to a tendency of overfitting the transition probability in this dataset. For all the methods, the performances drop from the monophonic to the polyphonic dataset, especially for BD and SD. This is an unsurprising trend. The less prominent decrease for HH might be due to the fact that the typical frequency range of HH is more separated from other instruments than BD and SD, thus is more robust against the presence of tonal sounds. In the case of template adaptation, a general trend of increase in precision and decrease in recall can be observed. One explanation is that once a better representation of the drum templates is found, the system might become more selective, leading toward a reduction in both false positives and true positives.

AM1 seems to perform better than AM2 on BD in both monophonic and polyphonic dataset. One possible explanation is that bass drum usually appears on the downbeats, which tends to have higher correlation with other entries in harmonic activation matrix. This means BD has a higher chance of being adapted to better templates using AM1. AM2 uses a more generalized adaptation process and performs better on HH and SD. However, it is more computationally demanding since it adapts the templates constantly, whereas AM1 only adapts when the correlation is above the threshold. To sum up, both template adaptation methods perform at the similar level, and the best fit of either method for specific types of music still needs to be investigated.

5. CONCLUSION

We have presented a drum transcription system for both monophonic and polyphonic music using partially fixed NMF with template adaptation. The system is robust against rank changes, and the evaluation results show that the two presented template adaptation methods improve the precision of the system, leading toward better performance. The proposed method is able to achieve average F-measures of 77.9% and 72.2% in monophonic and polyphonic music respectively for detecting 3 classes of drums.

The presented method has the following advantages:

Method	Metric	HH	BD	SD	Mean
PFNMF	P	0.918	0.886	0.825	0.876
	R	0.705	0.938	0.453	0.698
	F	0.797	0.911	0.585	0.764
AM1	P	0.909	0.955	0.837	0.900
	R	0.682	0.927	0.473	0.694
	F	0.779	0.940	0.604	0.774
AM2	P	0.928	0.914	0.854	0.898
	R	0.703	0.927	0.483	0.704
	F	0.799	0.920	0.617	0.779
Gillet et al. [9]	P	0.736	0.798	0.710	0.748
	R	0.865	0.700	0.642	0.735
	F	0.795	0.745	0.674	0.738
Paulus et al. [16]	P	0.838	0.941	0.750	0.806
	R	0.849	0.921	0.567	0.843
	F	0.843	0.930	0.645	0.779

Table 1. Evaluation results for ENST drum dataset *minus one* subset **without** accompaniments

Method	Metric	HH	BD	SD	Mean
PFNMF	P	0.902	0.714	0.684	0.766
	R	0.706	0.862	0.464	0.677
	F	0.792	0.781	0.552	0.708
AM1	P	0.904	0.781	0.758	0.814
	R	0.679	0.856	0.45	0.661
	F	0.775	0.816	0.564	0.719
AM2	P	0.908	0.774	0.726	0.802
	R	0.694	0.855	0.466	0.671
	F	0.786	0.812	0.567	0.722
Gillet et al. [9]	P	0.702	0.744	0.619	0.688
	R	0.818	0.653	0.552	0.674
	F	0.755	0.695	0.583	0.678
Paulus et al. [16]	P	0.847	0.802	0.663	0.770
	R	0.826	0.815	0.453	0.698
	F	0.836	0.808	0.538	0.727

Table 2. Evaluation results for ENST drum dataset *minus one* subset **with** accompaniments

First, the system only requires a few training samples for template extraction, and these templates can adapt toward the target sources gradually. This makes the system more applicable to the real world use case. Second, adjustment of the parameter r_H allows the algorithm to work with polyphonic music, and the use of a weighting matrix prevents the performance from dropping as r_H increases. Third, the cross-dataset evaluation results indicate a robustness against template mismatches, possibly allowing the application in situations with minimum prior knowledge. Last but not least, the evaluation results indicate that the F-measure of the proposed methods is at the same level as state-of-the-art systems with a lower model complexity.

Possible directions for future work include the automatic estimation of r_H for any given signal using a probabilistic approach similar to [19]; this might be a solution for the system to optimally select the rank. Furthermore, a more detailed analysis of playing techniques might be necessary toward a more complete drum transcription system. Finally, different penalty terms for the NMF cost function, such as sparsity, temporal continuity [24], or rank r_H might be taken into account for better adjustment of the current method.

6. REFERENCES

- [1] David S Alves, Jouni Paulus, and José Fonseca. Drum transcription from multichannel recordings with non-negative matrix factorization. In *Proc. of the European Signal Processing Conference (EUSIPCO)*, Glasgow, 2009.
- [2] Emmanouil Benetos, Simon Dixon, Dimitrios Gianoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: Challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, December 2013.
- [3] Emmanouil Benetos, Sebastian Ewert, and Tillman Weyde. Automatic transcription of pitched and unpitched sounds from polyphonic music. In *Proc. of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 2014.
- [4] Christian Dittmar. Drum detection from polyphonic audio via detailed analysis of the time frequency domain. In *Proc. of the Music Information Retrieval Evaluation eXchange (MIREX)*, 2005.
- [5] Christian Dittmar and Daniel Gärtner. Real-time Transcription and Separation of Drum Recording Based on NMF Decomposition. In *Proc. of the International Conference on Digital Audio Effects (DAFX)*, pages 1–8, 2014.
- [6] Derry FitzGerald, Bob Lawlor, and Eugene Coyle. Drum transcription in the presence of pitched instruments using prior subspace analysis. In *Proc. of the*

- Irish Signals & Systems Conference (ISSC)*, Limerick, 2003.
- [7] Olivier Gillet and Gaël Richard. Automatic transcription of drum loops. In *Proc. of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 4, pages 269–272, May 2004.
- [8] Olivier Gillet and Gaël Richard. ENST-Drums: an extensive audio-visual database for drum signals processing. In *Proc. of the International Conference on Music Information Retrieval (ISMIR)*, Victoria, 2006.
- [9] Olivier Gillet and Gaël Richard. Transcription and separation of drum signals from polyphonic music. *IEEE transactions on Audio, Speech, and Language Processing*, 16(3):529–540, March 2008.
- [10] Marko Helen and Tuomas Virtanen. Separation of drums from polyphonic music using non-negative matrix factorization and support vector machine. In *Proc. of the European Signal Processing Conference (EUSIPCO)*, Antalya, 2005.
- [11] Perfecto Herrera, Amaury Dehamel, and Fabien Gouyon. Automatic labeling of unpitched percussion sounds. In *Proc. of the 114th Audio Engineering Society Convention*. AES, March 2003.
- [12] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in Neural Information Processing Systems (NIPS)*, 2000.
- [13] Alexander Lerch. *An Introduction to Audio Content Analysis: Applications in Signal Processing and Music Informatics*. John Wiley & Sons, 2012.
- [14] Henry Lindsay-Smith, Skot McDonald, and Mark Sandler. Drumkit Transcription via Convolutional NMF. In *Proc. of the International Conference on Digital Audio Effects (DAFX)*, pages 15–18, 2012.
- [15] Arnaud Moreau and Arthur Flexer. Drum transcription in polyphonic music using non-negative matrix factorization. In *Proc. of the International Conference on Music Information Retrieval (ISMIR)*, pages 353–354, 2007.
- [16] Jouni Paulus and Anssi Klapuri. Drum Sound Detection in Polyphonic Music with Hidden Markov Models. *EURASIP Journal on Audio, Speech, and Music Processing*, pages 1–9, 2009.
- [17] Jouni Paulus and Tuomas Virtanen. Drum transcription with non-negative spectrogram factorisation. In *Proc. of the European Signal Processing Conference (EUSIPCO)*, page 4, Antalya, 2005.
- [18] Stanislaw A. Raczyski, Nobutaka Ono, and Shigeki Sagayama. Multipitch analysis with harmonic nonnegative matrix approximation. In *Proc. of International Conference on Music Information Retrieval (ISMIR)*, 2007.
- [19] Mikkel N. Schmidt and Morten Mørup. Infinite non-negative matrix factorization. In *Proc. of the European Signal Processing Conference (EUSIPCO)*, 2010.
- [20] Paris Smaragdis and Judith C Brown. Non-negative matrix factorization for polyphonic music transcription. In *Proc. of the Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, New Paltz, 2003. .
- [21] Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka. Supervised and semi-supervised separation of sounds from single-channel mixtures. In *Proc. of the 7th international conference on Independent component analysis and signal separation*, pages 414–421, 2007.
- [22] Koen Tanghe, Sven Degroove, and Bernard De Baets. An algorithm for detecting and labeling drum events in polyphonic music. In *Proc. of the Music Information Retrieval Evaluation eXchange (MIREX)*, 2005.
- [23] Lucas Thompson, Matthias Mauch, and Simon Dixon. Drum Transcription via Classification of Bar-Level Rhythmic Patterns. In *Proc. of the International Conference on Music Information Retrieval (ISMIR)*, 2014.
- [24] Tuomas Virtanen. Monaural sound source separation by nonnegative matrix factorization with temporal continuity and sparseness criteria. *IEEE transactions on audio, speech, and language processing*, 15(3):1066–1074, 2007.
- [25] Chih-Wei Wu and Alexander Lerch. Drum Transcription using Partially Fixed Non-Negative Matrix Factorization. In *Proc. of the European Signal Processing Conference (EUSIPCO)*, Nice, France, 2015.
- [26] Jiho Yoo, Minje Kim, Kyeongok Kang, and Seungjin Choi. Nonnegative matrix partial co-factorization for drum source separation. In *Proc. of the International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 1942–1945, Dallas, 2010. .
- [27] Kazuyoshi Yoshii, Masataka Goto, and Kazumori Komatani. Drumix: An audio player with real-time drum-part rearrangement functions for active music listening. *IPSI Digital Courier*, 3:134–144, 2007.
- [28] Kazuyoshi Yoshii, Masataka Goto, and Hiroshi G. Okuno. Automatic drum sound description for real-world music using template adaptation and matching methods. In *Proc. of the International Conference on Music Information Retrieval (ISMIR)*, Barcelona, 2004.
- [29] Kazuyoshi Yoshii, Masataka Goto, and Hiroshi G. Okuno. Drum sound recognition for polyphonic audio signals by adaptation and matching of spectrogram templates with harmonic structure suppression. *IEEE transactions on Audio, Speech and Language Processing*, 15(1):333–345, January 2007.

BEAT AND DOWNBEAT TRACKING BASED ON RHYTHMIC PATTERNS APPLIED TO THE URUGUAYAN CANDOMBE DRUMMING

Leonardo Nunes Martín Rocamora Luis Jure Luiz W. P. Biscainho
ATL-Brazil, Microsoft Universidad de la República Federal Univ. of Rio de Janeiro
lnunes@microsoft.com {rocamora,lj}@eumus.edu.uy wagner@smt.ufr.br

ABSTRACT

Computational analysis of the rhythmic/metrical structure of music from recorded audio is a hot research topic in music information retrieval. Recent research has explored the explicit modeling of characteristic rhythmic patterns as a way to improve upon existing beat-tracking algorithms, which typically fail on dealing with syncopated or poly-rhythmic music. This work takes the Uruguayan Candombe drumming (an afro-rooted rhythm from Latin America) as a case study. After analyzing the aspects that make this music genre troublesome for usual algorithmic approaches and describing its basic rhythmic patterns, the paper proposes a supervised scheme for rhythmic pattern tracking that aims at finding the metric structure from a Candombe recording, including beat and downbeat phases. Then it evaluates and compares the performance of the method with those of general-purpose beat-tracking algorithms through a set of experiments involving a database of annotated recordings totaling over two hours of audio. The results of this work reinforce the advantages of tracking rhythmic patterns (possibly learned from annotated music) when it comes to automatically following complex rhythms. A software implementation of the proposal as well as the annotated database utilized are available to the research community with the publication of this paper.

1. INTRODUCTION

Meter plays an essential role in our perceptual organization of music. In modern music theory, metrical structure is described as a regular pattern of points in time (*beats*), hierarchically organized in metrical levels of alternating strong and weak beats [15, 16]. The metrical structure itself is not present in the audio signal, but is rather inferred by the listener through a complex cognitive process. Therefore, a computational system for metrical analysis from audio signals must, explicit or implicitly, make important cognitive assumptions. A current cognitive model proposes that, given a temporal distribution of events, a

competent listener infers the appropriate metrical structure by applying two sets of rules: Metrical Well-Formedness Rules (MWFR), which define the set of possible metrical structures, and Metrical Preference Rules (MPR), which model the criteria by which the listener chooses the most stable metrical structure for a given temporal distribution of events [15]. While not strictly universal, most of the MWFR apply for a variety of metric musics of different cultures [23]; MPR, on the other hand, are more subjective and, above all, style-specific. A listener not familiar with a certain type of music may not be able to decode it properly, if its conventions differ substantially from usual tonal metrical structures.

This is why the computational analysis of rhythmic/metrical structure of music from audio signals remains a difficult task. Most generic algorithms follow a bottom-up approach with little prior knowledge of the music under analysis [6, 7, 13], often including some kind of preference rules—e.g. by aligning beats with onsets of stronger and/or longer events [15]. Therefore, they usually fail on processing syncopated or polyrhythmic music, for instance, that of certain Turkish, Indian or African traditions [22].

For this reason, other approaches prefer a top-down process guided by high-level information, such as style-specific characteristics [11]. Given that listeners tend to group musical events into recurrent rhythmic patterns which give cues for temporal synchronization, the explicit modeling of rhythmic patterns has recently been proposed as a way to improve upon existing beat-tracking algorithms [14, 24, 25]. The identification of challenging music styles and the development of style-specific algorithms for meter analysis and beat-tracking is a promising direction of research to overcome the limitations of existing techniques.

In this work, an afro-rooted rhythm is considered as a case of study: the Candombe drumming in Uruguay. Motivated by the fact that some characteristics of Candombe are challenging for most of the existing rhythm analysis algorithms, a supervised scheme for rhythmic pattern tracking is proposed, aiming at finding the metric structure from an audio signal, including the phase of beats and downbeats. The performance of the proposed method is assessed over a database of recordings annotated by an expert.

The next section provides a brief description of the Candombe rhythm. Then, the proposed method for rhythmic pattern matching is presented in Section 3. Experiments and results are described in Section 4. The paper ends with some critical discussion and directions for future research.



© Leonardo Nunes, Martín Rocamora, Luis Jure, Luiz W. P. Biscainho. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Leonardo Nunes, Martín Rocamora, Luis Jure, Luiz W. P. Biscainho. "Beat and Downbeat Tracking Based on Rhythmic Patterns Applied to the Uruguayan Candombe Drumming", 16th International Society for Music Information Retrieval Conference, 2015.

2. AFRO-URUGUAYAN CANDOMBE

2.1 Candombe drumming in context

Candombe is one of the most characteristic features of Uruguayan popular culture, practiced by thousands of people. Its rhythm influenced and was incorporated into various genres of popular music. However, little known abroad, it may be difficult to understand for unfamiliar listeners.

Although originated in Uruguay, Candombe has its roots in the culture brought by the African slaves in the 18th century. It evolved during a long historical process, gradually integrating European immigrants and now permeating the whole society [1, 8]. Candombe drumming, with its distinctive rhythm, is the essential component of this tradition. Its most characteristic manifestation is the *llamada de tambores*, a drum-call parade, when groups of players meet at specific points in the city to play while marching on the street (Figure 1).



Figure 1. Group of Candombe drummers.

The instrument of Candombe is called *tambor* (“drum” in Spanish), of which there are three different sizes: *chico* (small), *repique* (medium) and *piano* (big). Each type has a distinctive sound (from high to low frequency range) and its own specific rhythmic pattern. All three are played with one hand hitting the skin bare and the other with a stick, which is also used to hit the shell when playing the *clave* pattern. The minimal ensemble of drums (*cuerda de tambores*) must have at least one of each of the three drums; during a *llamada de tambores* it usually consists of around 20 to 60 drums. While marching, the players walk forward with short steps synchronized with the beat or *tactus*; this movement, while not audible, is very important for the embodiment of the rhythm. Figure 1 shows in the first row, from the front backwards, a *repique*, a *chico* and a *piano*.

2.2 Rhythmic patterns and metrical structure

The Candombe rhythm or *ritmo de llamada* results from the interaction between the patterns of the three drums. An additional important pattern is the *clave*, played by all the drums as an introduction to and preparation for the rhythm (see Figure 2¹).

The pattern of the *chico* drum is virtually immutable, and establishes the lowest level of the metrical structure

¹ Lower and upper line represent hand and stick strokes respectively.



Figure 2. Interaction of main *Candombe* patterns, and the three levels of the resulting metric structure. *Repique* and *piano* patterns are shown in a simplified basic form.

(*tatum*). The period of the pattern is four *tatums*, conforming the beat or *tactus* level in the range of about 110 to 150 beats per minute (BPM). The interaction of *chico* and *clave* helps to establish the location of the beat within the *chico* pattern (otherwise very difficult to perceive), and defines a higher metric level of four beats (sixteen *tatums*).

The resulting metrical structure is a very common one: a four-beat measure with a regular subdivision in 16 *tatums*. However, two characteristic traits link the rhythmic configuration of Candombe with the Afro-Atlantic music traditions, differentiating it from usual tonal rhythms: 1) the pattern defining the pulse does not articulate the *tatum* that falls on the beat, and has instead a strong accent on the second; 2) the *clave* divides the 16-*tatum* cycle irregularly (3+3+4+2+4), with only two of its five strokes coinciding with the beat. This makes the Candombe rhythm difficult to decode for both listeners not familiar with it and generic beat-tracking algorithms (see Table 1). The strong phenomenological accents displaced with respect to the metric structure add to the difficulty.

The *repique* is the drum with the greatest degree of freedom. During the *llamada* it alternates between the *clave* pattern and characteristically syncopated phrases. Figure 2 shows its primary pattern, usually varied and improvised upon to generate phrases of high rhythmic complexity [12]. The *piano* drum has two functions: playing the base rhythm (*piano base*), and occasional more complex figurations akin to the *repique* phrases (*piano repicado*). The pattern in Figure 2 is a highly abstracted simplification of the *piano base*. It can be seen that it is essentially congruent with the *clave* pattern, and when correctly decoded it permits the inference of the whole metric structure. In real performances, however, much more complex and varied versions of this pattern are played. It has been shown [21] that the analysis of *piano* patterns may elicit the identity of different neighborhoods (*barrios*)² and individual players.

3. RHYTHMIC PATTERN MATCHING

In this section, a rhythmic/metric analysis algorithm that matches a given rhythmic accentuation pattern to an audio signal is described. It tries to find the time of occurrence

² The three more important traditional styles are *Cuareim* (or *barrio Sur*), *Ansina* (or *barrio Palermo*) and *Gaboto* (or *barrio Cordón*).

of each *tatum* knowing its expected accentuation inside the pattern, thus being able to track not only the beat but also other metrical information. Initially, a tempo estimation algorithm is employed to obtain the beat period (tempo), assumed to be approximately stable throughout the signal. Then, the main algorithm is used to find the phase of the accentuation pattern within the observed signal.

3.1 Audio feature extraction

For audio feature extraction, this work adopts a typical approach based on the Spectral Flux. First, the Short-Time Fourier Transform of the signal is computed and mapped to the MEL scale for sequential windows of 20 ms duration in hops of 10 ms. The resulting sequences are differentiated (via first-order difference) and half-wave rectified.

For tempo estimation, the feature values are summed along all MEL sub-bands, in order to take into account events from any frequency range.

Since its pattern is the most informative on both *tactus* beat and downbeat locations, the rhythmic pattern tracking is tailored towards the *piano* (i.e. the lowest) drum. Therefore, the accentuation feature used for pattern matching is obtained by summing the Spectral Flux along the lowest MEL sub-bands (up to around 200 Hz) only. This function is normalized by the 8-norm of a vector containing its values along ± 2 estimated *tatum* periods around the current frame. The resulting feature value is expected to be close to one if a pulse has been articulated and close to zero otherwise. In addition, it also carries some information on the type of articulation. For instance, an accented stroke produces a higher feature value compared to a muffled one, since in the former case the spectral change is more abrupt.

3.2 Tempo Estimation

For tempo estimation, this work adopts a straightforward procedure based on locating the maximum of a suitably defined similarity function. As proposed in [20], the basic function is the product between the auto-correlation function and the Discrete Fourier Transform of the features computed for the whole signal. The result is weighted by the function described in [17]. The period associated with the largest value in this weighted similarity function is selected as the tempo of the signal. After the tempo is obtained, the *tatum* period used for pattern tracking can be computed just by dividing the beat period by 4. This *tatum* period is then used to define the variables in the pattern tracking algorithm as described in the next sections.

3.3 Variables definition

In order to perform its task, the algorithm employs two discrete random variables. The first one, called *tatum counter*, c_k , counts how many frames have passed since the last *tatum* has been observed at frame k . Assuming an estimated *tatum* period of τ frames, then $c_k \in \{0, 1, \dots, \tau - 1 + \sigma_c\}$, where σ_c is a parameter that allows for possible timing inaccuracies in the *tatum*. The second, called *pattern index*, a_k , indicates the position inside a given rhythmic

mic pattern at frame k in the range $\{0, 1, \dots, M - 1\}$, where M is the length of the rhythmic pattern in *tatums*. The rhythmic pattern will be expected to define a series of accents or lacks of accent in the *tatums*. Time evolution of these two variables will be described in the next section, where it is assumed that the sampling rate of the feature (typically less than 100 Hz) is much lower than that of the original signal (usually 44.1 kHz). The model describes the accentuation feature extracted at frame k as a random variable, y_k , with actual observed (extracted) value y_k .

3.4 State Transition

In this section, the probabilities of each value for the two random variables at frame k given past frames are described. A first-order Markov model will be assumed for the joint distribution of the random variables, i.e., the probability of each possible value of a random variable at frame k depends only on the values assumed by the variables at the previous frame $k - 1$. Using this assumption, the two random variables will constitute a Hidden Markov Model [18].

The *tatum* counter variable, as previously mentioned, counts how many frames have passed since the last *tatum*. The state $c_k = 0$ is considered the “*tatum state*” and indicates that a *tatum* has occurred at frame k . This random variable is closely related to the *phase state* proposed in [5] for beat tracking. Only two possible transitions from frame $k - 1$ to frame k are allowed: a transition to the “*tatum state*” or an increment in the variable. The transition to the “*tatum state*” depends on both the past value of the variable and the (known) *tatum* period. The closer the value of the variable is to the *tatum* period, the more probable is the transition to the “*tatum state*.” Mathematically, it is possible to write

$$p_{c_k}(c_k | c_{k-1}) = \begin{cases} h[c_{k-1} - \tau], & \text{if } c_k = 0 \\ 1 - h[c_{k-1} - \tau], & \text{if } c_k = c_{k-1} + 1 \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

where $h[\cdot]$ is a tapering window with $h[n] = 0$ for $|n| > \sigma_c$ that models possible timing inaccuracies on the *tatum*, and $\sum_n h[n] = 1$. Currently, a normalized Hann window is employed to penalize farther values. The value $\sigma_c = 2$ was set for the reported experiments, indicating that inaccuracies of up to 50 ms are tolerated by the algorithm.

Since the accentuation pattern is defined in terms of the *tatum*, its time evolution will be conditioned by the pattern evolution. Assuming that the pattern indicates the expected accentuation of the next *tatum*, the variable should only change value when a “*tatum state*” has been observed, indicating that a different accentuation should be employed by the observation model (described in the next section). Hence, mathematically

$$p_{a_k}(a_k | c_{k-1}, a_{k-1}) = \begin{cases} 1, & \text{if } (a_k = a_{k-1} \oplus 1) \wedge (c_{k-1} = 0) \\ 1, & \text{if } (a_k = a_{k-1}) \wedge (c_{k-1} \neq 0) \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

where \wedge is the logical AND, \oplus denotes a modulo- M summation, and M is the length of the accentuation pattern. As can be gathered, given the previous *tatum* counter value, the pattern index becomes deterministic, with its next value completely determined by its value at the previous frame and the value of the *tatum* counter. The transitions for this variable are inspired on the ones used in the family of algorithms based on [24] (i.e. [2, 10, 14]), except for defining the pattern in terms of *tatums* instead of an arbitrary unit.

3.5 Observation Model

This section describes the likelihood of c_k and a_k given an observed accentuation y_k in the signal. The main idea is to measure the difference between the expected accentuation (provided by the rhythmic pattern) and the observed one. The larger the difference, the less probable the observation.

If the accentuation pattern is a vector $\bar{A} \in \mathbb{R}^{M \times 1}$ containing the expected feature values, then at frame k the likelihood for $c_k = 0$ (“tatum state”) can be defined as

$$p_{y_k}(y_k | c_k, a_k) = N_{\sigma_c}(y_k - \bar{A}_{a_k}), \quad (3)$$

where $N_{\sigma_c}(\cdot)$ is a Gaussian function with zero mean and variance σ_c^2 used to model possible deviations between expected and observed accents. For $c_k \neq 0$, the likelihood is given by:

$$p_{y_k}(y_k | c_k, a_k) = N_{\sigma_d}(y_k), \quad (4)$$

where N_{σ_d} is a zero-mean Gaussian with variance equal to σ_d^2 . Hence, the closer to zero the feature, the more probable the observation. This is similar to the non-beat model adopted in [5], and is not found in [14, 24].

In the reported experiments, $\sigma_c = \sigma_d = 0.5$, thus allowing for a reasonable overlap between expected and actual observed values.

3.6 Inference

A summary of the proposed model for rhythmic pattern tracking can be viewed in Figure 3, where the statistical dependencies among the variables are explicated. Different inference strategies can be employed to find the most probable pattern index and *tatum* counter values given the observed accentuation [18]. In this work, the well-known Viterbi algorithm [18, 24] is employed to find the most probable path among all possible combinations of values of each random variable given the observed features y_k .

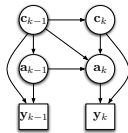


Figure 3. Graphical representation of the statistical dependency between random variables and observations.

At last, a uniform prior is chosen for c_0 and a_0 indicating that the counter and the pattern can start with any possible value in the first frame.

4. EXPERIMENTS AND RESULTS

A set of experiments was devised to assess the performance of the proposed rhythmic pattern tracking system with respect to the problems of estimating the rate and phase of beats and downbeats, using a database of manually labeled Candombe recordings. Four state-of-the-art beat-tracking algorithms [6, 7, 13, 19] were included in the experiments in order to evaluate how challenging the rhythm at hand is for typical general-purpose approaches.

Two different strategies are explored: the rhythmic patterns to follow are either informed to the algorithm based on a priori musical knowledge about the rhythm, or learned from the labeled database itself.

4.1 Dataset

A dataset of Candombe recordings, totaling over 2 hours of audio, was compiled and annotated for this work and it is now released to the research community.³ It comprises 35 complete performances by renowned players, in groups of three to five drums. Recording sessions were conducted in studio, in the context of musicological research over the past two decades. A total of 26 *tambor* players took part, belonging to different generations and representing all the important traditional Candombe styles. The audio files are stereo with a sampling rate of 44.1 kHz and 16-bit precision. The location of beats and downbeats was annotated by an expert, adding to more than 4700 downbeats.

4.2 Performance measures

Since tempo estimation is only an initialization step of the rhythmic pattern tracking task, whose overall performance will be examined in detail, it suffices to mention that the estimated tempo was within the interval spanned by the annotated beat periods along each of the files in the database, thus providing a suitable value for the respective variable.

Among the several objective evaluation measures available for audio beat tracking [4] there is currently no consensus over which to use, and multiple accuracies are usually reported [2, 3]. In a recent pilot study, the highest correlation between human judgements of beat tracking performance and objective accuracy scores was attained for CMLT and Information Gain [3].

In this work CMLT, AMLT and F-measure were adopted, as their properties are well understood and were considered the most suitable for the current experiments. The non-inclusion of Information Gain was based on the observation that it yielded high score values for estimated beat sequences that were definitely not valid. Specifically, in several instances when the beat rate (or a multiple of it) was precisely estimated, even if the beat phase was repeatedly misidentified, the Information Gain attained high values while other measures such as CMLT or F-measure were coherently small. In the following, a brief description

³ Available from <http://www.eumus.edu.uy/candombe/datasets/ISMIR2015/>.

of the adopted metrics⁴ is provided (see [4] for details), along with the values selected for their parameters.

The CMLt measure (Correct Metrical Level, continuity not required) considers a beat correctly estimated if its time-difference to the annotated beat is below a small threshold, and if the same holds for the previous estimated beat. Besides, the inter-beat-interval is required to be close enough to the inter-annotation-interval using another threshold. The total number of correctly detected beats is then divided by the number of annotated beats and expressed as a percentage (0-100%). Both thresholds are usually set to 17.5% of the inter-annotated-interval, which was also the value adopted in this work. The AMLt measure (Allowed Metrical Levels, continuity not required) is the same as CMLt but does not take into account errors in the metrical level and phase errors of half the period.

The F-measure (Fmea) is the harmonic mean of precision and recall of correctly detected beats, where precision stands for the ratio between correctly detected beats and the total number of estimated beats, while recall denotes the ratio between correctly detected beats and the total number of annotated beats. A beat is considered correctly detected if its time-difference to the annotation is within ± 70 ms; this tolerance was kept in this work.

Only CMLt and F-measure were used for assessing the downbeat, since the loosening of metrical level and phase constraints in AMLt was considered inappropriate.

4.3 Experiments with informed rhythmic patterns

In the first type of experiment, the pattern to track \bar{A} is informed to the algorithm based on musical knowledge about the rhythm, without any training or tuning to data. On one hand, this has a practical motivation: even when no labeled data is available one could take advantage of the technique. On the other hand, it gives a framework in which musical models can be empirically tested. In short, an informed rhythmic pattern based on musical knowledge is nothing but a theoretical abstraction, and this type of experiment could provide some evidence of its validity.

To that end, based on the different ways the *piano* pattern is notated by musicology experts, a straightforward approach was adopted. Firstly, the *piano* pattern as introduced in Figure 2 (usually regarded as the *piano* in its minimal form) was considered. A binary pattern \bar{A} was assembled by setting a value of 1 for those *tatums* which are expected to be articulated and 0 otherwise. Then, a more complex pattern was considered by adding two of the most relevant articulated *tatums* which were missing, namely the 6th and 15th, and also building the corresponding binary pattern. Hence, the binary informed patterns proposed are Pattern 1: $\bar{A} = [1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 1, 1, 0, 0, 0]$ Pattern 2: $\bar{A} = [1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 0, 1, 1, 0, 1, 0]$.

This is certainly an oversimplification of the real rhythmic patterns, since it does not take into account the accented and muffled strokes that are an essential trait of a

piano performance. It would be possible to encompass dynamic variations into the informed pattern by considering distinct quantized values of the feature for different type of strokes. However, the binary patterns were favoured for the sake of simplicity and as a proof of concept.

Table 1⁵ compares 4 general-purpose beat-tracking algorithms with the proposed algorithm using the binary informed patterns, and also (for conciseness) the experiments discussed in the next section. Results are averaged over the whole database and weighted by the number of beats and downbeats of each audio file. Although the beat rate (or a multiple) is sometimes precisely estimated by the general-purpose beat-tracking algorithms, the correct metrical level and/or the phase of the beat is usually misidentified.

	BEAT			DOWNBEAT	
	CMLt	AMLt	Fmea	CMLt	Fmea
General-purpose					
Ellis [7]	44.2	63.0	43.8	-	-
Dixon [6]	13.9	14.9	22.7	-	-
IBT [19]	9.1	27.6	16.7	-	-
Klapuri [13]	28.8	35.5	29.3	36.6	13.2
Informed patterns – Section 4.3					
Pattern 1	80.2	80.5	81.3	84.7	79.1
Pattern 2	79.0	81.0	79.8	81.2	77.5
Learned patterns – Section 4.4 (leave-one-out)					
Median	79.9	79.9	80.8	82.4	76.9
K-means 2	81.7	81.7	82.6	84.4	79.3
K-means 5	82.5	82.5	83.6	85.2	80.6

Table 1. Performance of the different algorithms considered.

4.4 Experiments with learned rhythmic patterns

The labeled database allows the study of the rhythmic patterns actually present in real performances. There are different possible approaches to extract a single rhythmic pattern to track from the annotated data. For each *tatum*-grid position in the bar-length pattern, all the feature values in the dataset can be collected, and their distribution can be modeled, e.g. by a GMM as in [14]. The distribution of feature values in the low-frequency range will be dominated by the *base* patterns of the *piano* drum, albeit there will be a considerable amount of *repicado* patterns [21]. In order to cope with that, a simple model was chosen: the median of feature values for each *tatum* beat, which is less influenced by outliers than the mean.

The problem with the median pattern is that it models different beat positions independently. A better suited approach is to group the patterns based on their similarity into a given number of clusters, and select the centroid of the majority cluster as a good prototype of the *base* pattern. This was applied in [21] to identify *base* patterns

⁴ Computed with standard settings using code at <https://code.soundsoftware.ac.uk/projects/beat-evaluation/>.

⁵ Additional details can be found in <http://www.eumus.edu.uy/candombe/papers/ISMIR2015/>.

of the *piano* drum in a performance, and similarly in [10] to learn rhythmic patterns from annotated data to adapt a beat-tracking model to specific music styles. Figure 4 shows the patterns learned from the whole database, using the median and the centroid of the majority cluster obtained with K-means for 2 and 5 clusters. It is remarkable that the differently learned patterns are quite similar, exhibiting the syncopated 4th *tatum* beat as the most accented one. The locations of articulated beats for the informed patterns of the previous section are also depicted, and are consistent with the learned patterns. The K-means approach turned out to be little sensitive to the number of clusters, yielding similar patterns from 1 to 6.

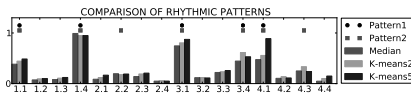


Figure 4. Comparison of the different patterns considered. (Median and K-means learned from the whole database.)

For testing the performance of the learning approach a leave-one-out scheme was implemented and the results are detailed in Table 1. Not surprisingly, performance is almost the same for the different rhythmic patterns. Considering different feature values instead of binary patterns did not yield any notable performance increase.

A detailed inspection of the performance attained for each recording in the database, as depicted in Figure 5, shows there is still some room for improvement, given that about half-a-dozen files are definitely mistracked. This may indicate that the pattern \bar{A} to track simply does not properly match the given performance. To check this hypothesis, a K-means ($K=2$) clustering was carried out only with the candidate patterns found within each target recording, whose tracking was then performed using the centroid of the majority cluster as \bar{A} . Table 2 shows the new results obtained for the files with lower performance (CMLT<50%) in the dataset. Except for the first one, performance was (sometimes notably) improved when the informed rhythmic pattern is the one that better matches the recording. Therefore, modeling several rhythmic patterns as in [10] can potentially improve the current results.

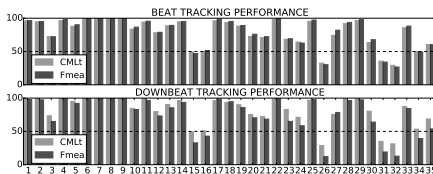


Figure 5. Leave-one-out performance for each recording of the database using the K-means pattern with $K=2$.

Recording #	BEAT		DOWNBEAT	
	CMLT	Fmea	CMLT	Fmea
15	34.1	32.8	32.7	7.2
16	95.6	98.0	96.3	97.1
26	40.2	36.9	42.9	22.2
31	71.3	69.9	78.3	67.8
32	55.7	54.1	59.6	44.7
34	60.9	60.0	62.7	51.7

Table 2. Scores attained when tracking the centroid of the majority cluster for each of the low performing files.

5. DISCUSSION AND FUTURE WORK

This paper tackled the problem of automatic rhythmic analysis of Candombe audio signals. A study of the rhythmic structure of Candombe was described, along with a pattern tracking algorithm that could deal with the particular characteristics of this rhythm. From the rhythm description and the presented experiments, it becomes clear that typical assumptions of general-purpose beat-tracking algorithms (such as strong events at beat times) do not hold, which hinders their performance. In order to overcome this problem, the proposed algorithm tracks a rhythmic pattern that informs when a beat with or without accentuation is expected to occur, which eventually can determine the complete metric structure. Indeed, experiments employing both rhythmic patterns based on musical knowledge and others learned from a labeled database, showed that the proposed algorithm can estimate the beat and downbeat positions for Candombe whereas traditional methods fail at these tasks. The attained CMLT score of about 80 % for beat tracking is approximately what one can expect from a state-of-the-art algorithm in a standard dataset [2,9], and what is reported in [10] for a Bayesian approach adapted to a culturally diverse music corpus. The present work gives additional evidence of the generalizability of the Bayesian approach to complex rhythms from different music traditions. The analysis of examples with low performance scores indicates that tracking several rhythmic patterns simultaneously, as proposed in [10], is a promising alternative for future work. Surely taking into account the timbre characteristics of different drums can be profitable.

Along with the annotated database employed, a software implementation of the proposal is being released with the publication of this paper to foster reproducible research (the first available implementation of the Bayesian approach for beat tracking, to the best of our knowledge).⁶

6. ACKNOWLEDGMENTS

This work was supported by funding agencies CAPES and CNPq from Brazil, and ANII and CSIC from Uruguay.

⁶ Available from github.com/lonnes/RhythmicAnalysis.

7. REFERENCES

- [1] G. Andrews. *Blackness in the White Nation: A History of Afro-Uruguay*. The University of North Carolina Press, Chapel Hill, USA, 2010.
- [2] S. Böck, F. Krebs, and G. Widmer. A multi-model approach to beat tracking considering heterogeneous music styles. In *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 603–608, Taipei, Taiwan, Oct. 2014.
- [3] M. Davies and S. Böck. Evaluating the evaluation measures for beat tracking. In *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 637–642, Taipei, Taiwan, Oct. 2014.
- [4] M. Davies, N. Degara, and M. Plumbley. Evaluation methods for musical audio beat tracking algorithms. Technical Report C4DM-TR-09-06, Center for Digital Music, Queen Mary University, London, UK, Oct. 2009.
- [5] N. Degara, E. A. Rúa, A. Pena, S. Torres-Guijarro, M. E. P. Davies, and M. D. Plumbley. Reliability-informed beat tracking of musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):290–301, Jan. 2012.
- [6] S. Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30(1):39–58, Aug. 2001.
- [7] D. P. W. Ellis. Beat tracking by dynamic programming. *Journal of New Music Research*, 36(1):51–60, May 2007.
- [8] L. Ferreira. An afrocentric approach to musical performance in the black south atlantic: The candombe drumming in Uruguay. *TRANS-Transcultural Music Review*, 11:1–15, Jul. 2007.
- [9] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon. Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(9):2539–2548, Nov. 2012.
- [10] A. Holzapfel, F. Krebs, and A. Rinivasamurthy. Tracking the “odd”: Meter inference in a culturally diverse music corpus. In *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 425–430, Taipei, Taiwan, Oct. 2014.
- [11] T. Jehan. *Creating Music by Listening*. PhD thesis, Program in Media Arts and Sciences, Massachusetts Institute of Technology, Cambridge, USA, Sep. 2005.
- [12] L. Jure. Principios generativos del toque de repique del candombe. In C. Aharonián, editor, *La música entre África y América*, pages 263–291. Centro Nacional de Documentación Musical Lauro Ayestarán, Montevideo, 2013. In Spanish.
- [13] A. P. Klapuri, A. J. Eronen, and J. T. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):342–355, Jan. 2006.
- [14] F. Krebs, S. Bck, and G. Widmer. Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *Proc. of the 14th International Conference on Music Information Retrieval (ISMIR 2013)*, pages 227–232, Curitiba, Brazil, Nov. 2013.
- [15] F. Lerdahl and R. Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, Cambridge, USA, 1985.
- [16] J. London. Cognitive constraints on metric systems: Some observations and hypotheses. *Music Perception*, 19(4):529–550, Summer 2002.
- [17] M. F. McKinney and D. Moelants. Deviations from the resonance theory of tempo induction. In *Proc. of the Conference on Interdisciplinary Musicology (CIM04)*, pages 1–11, Graz, Austria, Apr. 2004.
- [18] K. P. Murphy. *Dynamic Bayesian Networks: Representation, Inference and Learning*. PhD thesis, Computer Science Division, University of California, Berkeley, Berkeley, USA, Fall 2002.
- [19] J. L. Oliveira, M. E. P. Davies, F. Gouyon, and L. P. Reis. Beat tracking for multiple applications: A multi-agent system architecture with state recovery. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(10):2696–2706, Dec. 2012.
- [20] Geoffroy Peeters. Template-based estimation of time-varying tempo. *EURASIP Journal on Advances in Signal Processing*, 2007(67215):1–14, Dec. 2006.
- [21] M. Rocamora, L. Jure, and L. W. P. Biscainho. Tools for detection and classification of piano drum patterns from candombe recordings. In *Proc. of the 9th Conference on Interdisciplinary Musicology (CIM14)*, pages 382–387, Berlin, Germany, Dec. 2014.
- [22] A. Srinivasamurthy, A. Holzapfel, and X. Serra. In search of automatic rhythm analysis methods for turkish and indian art music. *Journal of New Music Research*, 43:94–114, Mar. 2014.
- [23] D. Temperley. *The Cognition of Basic Musical Structures*. MIT Press, Cambridge, USA, 2001.
- [24] N. Whiteley, A. Cemgil, and S. Godsill. Bayesian modelling of temporal structure in musical audio. In *Proc. of the 7th International Conference on Music Information Retrieval (ISMIR 2006)*, pages 29–34, Victoria, Canada, Oct. 2006.
- [25] M. Wright, W. A. Schloss, and G. Tzanetakis. Analyzing afro-cuban rhythms using rotation-aware clave template matching with dynamic programming. In *Proc. of the 9th International Conference on Music Information Retrieval (ISMIR 2008)*, pages 647–652, Philadelphia, USA, Sep. 2008.

AUTOMATED ESTIMATION OF RIDE CYMBAL SWING RATIOS IN JAZZ RECORDINGS

Christian Dittmar
International Audio
Laboratories Erlangen
christian.dittmar
@audiolabs-erlangen.de

Martin Pfeleiderer
The Liszt University of Music Weimar
martin.pfleiderer
@hfm-weimar.de

Meinard Müller
International Audio
Laboratories Erlangen
meinard.mueller
@audiolabs-erlangen.de

ABSTRACT

In this paper, we propose a new method suitable for the automatic analysis of microtiming played by drummers in jazz recordings. Specifically, we aim to estimate the drummers' swing ratio in excerpts of jazz recordings taken from the Weimar Jazz Database. A first approach is based on automatic detection of ride cymbal (RC) onsets and evaluation of relative time intervals between them. However, small errors in the onset detection propagate considerably into the swing ratio estimates. As our main technical contribution, we propose to use the log-lag autocorrelation function (LLACF) as a mid-level representation for estimating swing ratios, circumventing the error-prone detection of RC onsets. In our experiments, the LLACF-based swing ratio estimates prove to be more reliable than the ones based on RC onset detection. Therefore, the LLACF seems to be the method of choice to process large amounts of jazz recordings. Finally, we indicate some implications of our method for microtiming studies in jazz research.

1 Introduction

Jazz drummers usually keep time by using the ride cymbal (RC) and hi-hat (HH), especially in styles with so-called "swing feel" [2]. They commonly emphasize the "backbeat," i.e., the metric-harmonically unaccented beat, on the HH while playing typical patterns on the RC. According to [21, p. 248], this supports the "light" character of jazz rhythm. Instead of playing the beat in a steady manner, variations and additional "offbeat" strokes are usually added on the RC as well as on other drum parts. These variations differ from drummer to drummer and from performance to performance [2, pp. 617-629].

The most common time-keeping pattern played on the RC is shown in Figure 1. In addition to conventional drum notation in the top row, we show a corresponding time-domain signal at 240 BPM with overlaid amplitude envelope (bold black curve) and the so-called novelty curve

(thin black curve). We color-code the relevant beats and subdivisions thereof as follows. The sequence starts with the so-called "downbeat" quarter note (light blue), followed by the backbeat eighth note (light green), and the offbeat eighth note (light red) before starting over again with the downbeat. We will refer to this prototype sequence of onsets as RC pattern.

The so-called swing ratio expresses the beat subdivision and relates to the phrasing of the eighth notes in the RC pattern. Swinging eighth notes are typically played in different ratios, ranging continuously from straight eighths (1 : 1), over triplet eighths (2 : 1), to dotted eighths (3 : 1), or more extreme ratios. The swing ratio is reported to be tempo dependent [4, 9, 15], cf. Section 2.1. In Figure 1, the color-coded tone durations show how the backbeat duration grows with increasing swing factor, while the complementing offbeat duration shrinks. In Figure 1(a), backbeat and offbeat have equal duration, corresponding to straight eighths as given in the drum notation. In Figure 1(b), the RC pattern is notated as tied-triplets. In Figure 1(c), the backbeat duration equals a dotted eighth. Consequently, the offbeat duration equals that of a sixteenth note as shown in the drum notation.

There are several case studies concerning the swing ratio in jazz (cf. [21, pp. 262-273], and Section 2.1). While most of the studies examine swing ratios of soloists, it is widely acknowledged that the swing ratio of the RC pattern crucially contributes to the "swinging" character of the music. Most of the studies are based on manual transcription of onsets, often by visual inspection of the amplitude envelope of jazz excerpts. Few studies specifically examine the RC pattern [15] and its interaction with the soloist's timing [9]. This inspired us to develop and to evaluate methods for automated swing ratio estimation from RC patterns in jazz recordings. For sure, an automated generation of large amounts of reliable swing ratio data is essential for meaningful and more differentiated research on microtiming in jazz. Besides onset-based swing ratio estimation, our main approach is a log-lag variant of a local autocorrelation function (ACF) applied to onset-related novelty functions (see Sections 3.3). We refer to this representation as log-lag ACF (LLACF) and show its applicability to swing ratio estimation in Section 3.4.



© Christian Dittmar, Martin Pfeleiderer, Meinard Müller.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Christian Dittmar, Martin Pfeleiderer, Meinard Müller. "Automated Estimation of Ride Cymbal Swing Ratios in Jazz Recordings", 16th International Society for Music Information Retrieval Conference, 2015.

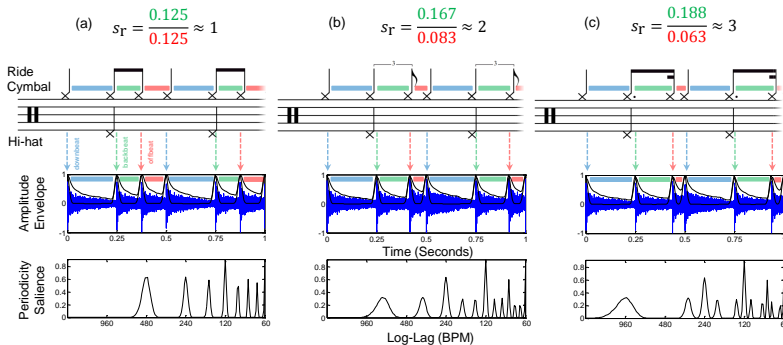


Figure 1. Illustration of prototypical RC patterns as drum notation (top), time-domain signal (mid), and LLACF (bottom). (a): Swing factor of $s_r = 1$ corresponding to straight eighth notes. (b): Swing factor of $s_r = 2$ corresponding to the idealized “tied-triplet feel”. (c): Swing factor $s_r = 3$, where the duration of the backbeat equals a dotted eighth note.

2 Related Work

A number of papers are concerned with systematic studies on swing ratio in jazz music. Since most of the studies use comparably small data sets and manual annotation, we think that swing ratio estimation is a suitable task to apply automatic methods from Music Information Retrieval (MIR) research in order to enable analysis of larger music data sets.

2.1 Jazz Microtiming Analysis

An early attempt to analyze swing ratios in jazz recordings is described in [17]. The author relies on visual inspection of spectrograms but does not report quantitative results. In [22], the swing ratios in the analyzed jazz recordings are reported to range from 1.48 to 1.82. Rose [23] reports an average swing ratio of 2.38 measured from amplitude envelopes. In [7], an average swing ratio of 1.75 is measured using a MIDI wind controller played by saxophonists. In [19], the analysis focuses on the RC and swing factors between 1.0 and 3.3 are reported without detailing the measurement method. In [6], an average swing ratio of 1.6 is measured using amplitude envelopes. Friberg and Sundström [9] annotated RC onsets in spectrograms of jazz excerpts. They report trends indicating a high negative correlation between the tempo and the swing ratio which seems to be valid across different drummers. In [3], an average swing ratio of 2.45 is measured in the performances of pianists playing a MIDI piano. In [1], comparably low swing ratios in the range between 0.9 to 1.7 are measured from amplitude envelopes. Honing and de Haas [15] conducted experiments with professional jazz drummers performing on a MIDI drum kit. Besides further evidence for the tempo dependency of swing ratios, the results show that jazz drummers have enormous control over their timing.

2.2 Rhythmic Mid-Level Features

Motivated by the need to design specialized mid-level features for music similarity estimation, several authors proposed conceptually similar, tempo-independent representations of rhythmic patterns. The basic observation is, that rhythmic patterns that are perceived as similar by human listeners may not be judged as similar by automatic methods. One of the main reasons is that the patterns are typically played in different tempi, which makes them unsuited for direct comparison. Therefore, Peeters [20] used tempo normalized spectral rhythm patterns to automatically classify ballroom dance styles. Holzapfel and Stylianou [13, 14] proposed to apply the scale transform to periodicity spectra to enable the use of conventional distance measures between rhythmic patterns despite tempo differences. Around the same time, the LLACF was proposed in [12] as well as the tempo-insensitive representation used for classification of ballroom dances in [16]. The LLACF was reported to be favorable over the scale transform for classification of Latin American rhythm patterns in [24]. The tempoigram as described in [11] is based on similar ideas and additionally features a cyclic post-processing to remedy the problem of octave ambiguity. Marchand and Peeters [18] revisited the scale transform and applied it to modulation spectra as tempo-independent feature, again for classification of ballroom dances. Eppler et al. [8] used peak ratios in the LLACF as features for detecting the swing feel but did not explicitly try to estimate swing ratios.

3 Method

In this section, we describe our approaches to automatic swing ratio estimation from excerpts of jazz recordings with swing feel. The first variant relies on peak-picking in an onset-related novelty curve (Section 3.1). The sec-

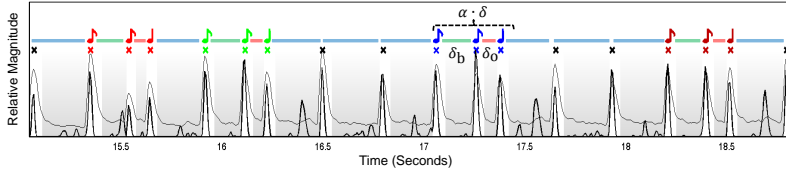


Figure 2. A four seconds excerpt from the 1979 recording of “Anthropology”, performed by Art Pepper playing solo clarinet, with Charlie Haden on bass and Billy Higgins on drums. The bold black curve depicts the novelty function Δ , the thin black curve shows the RC related threshold H . Automatically detected RC onsets are marked by the bold black crosses, colored crosses represent the four onset triples accepted for swing ratio estimation. The note durations are color-coded in the same way as in Figure 1.

ond approach relies on computation of the LLACF from the novelty curve (Section 3.3) and comparison to prototype LLACFs. As will be explained in Section 4.1, we have a rough tempo estimate $\tau_e \in \mathbb{R}_{>0}$ available for each jazz excerpt. Let $\delta_b, \delta_o \in \mathbb{R}_{>0}$ be the tone duration of the backbeat and the offbeat in an RC pattern as shown in Figure 1. They relate to the tempo by $\tau_e \approx (\delta_b + \delta_o)^{-1} \approx \delta^{-1}$, with the beat (quarter note) duration $\delta \in \mathbb{R}_{>0}$. The targeted swing ratio is given by:

$$s_r = \frac{\delta_b}{\delta_o} \quad (1)$$

Consequently, $\delta_b = \delta \cdot s_r \cdot (1 + s_r)^{-1}$ yields the tone duration of the backbeat and $\delta_o = \delta \cdot (1 + s_r)^{-1}$ yields the tone duration of the offbeat.

3.1 Ride Cymbal Onset Detection

With regard to Eqn (1), we aim to measure δ_b and δ_o from the jazz excerpts under analysis. One possibility is to search for RC onsets and use the time differences between consecutive onsets as estimate for note durations. To this end, we compute a time-frequency (TF) representation of an excerpt using the short-time Fourier transform (STFT) with blocksize w and hopsize r given in seconds. Let $\mathcal{X}(m, k)$ with $m \in [1 : M]$, $k \in [0 : K]$ be a complex-valued STFT coefficient at the m^{th} time frame and k^{th} spectral bin. Here, the interval $[1 : M]$ represents the time axis and K corresponds to the Nyquist frequency. Following the approaches in [10, 11], we compute a novelty curve $\Delta : [1 : M] \rightarrow \mathbb{R}$ as follows. First, we derive the logarithmically compressed magnitude spectrogram $\mathcal{Y}(m, k) := \log(1 + \gamma \cdot |\mathcal{X}(m, k)|)$ for a suitable constant $\gamma \geq 1$. Then, the novelty function is given as

$$\Delta(m) := \sum_{k=0}^K |\mathcal{Y}(m+1, k) - \mathcal{Y}(m, k)|_{\geq 0}, \quad (2)$$

where $|\cdot|_{\geq 0}$ denotes half-wave rectification. The resulting Δ exhibits salient peaks at frames corresponding to tone onsets. Inevitably, spurious peaks may occur in Δ that could be mistaken for RC onsets. Thus, we derive an RC

related threshold function as

$$H(m) := \sum_{k=k_0}^K |\mathcal{X}(m, k)|, \quad (3)$$

where the bin k_0 corresponds to the lower cutoff frequency. Figure 2 shows an example of Δ as bold black curve and the corresponding H as thin black curve. For the sake of visibility, both curves are normalized to unit maximum in the plot. We take the average value of H as threshold criterion and only accept peaks from Δ in frames where H exceeds this value (indicated by the white background). The $N = 18$ local maxima accepted as RC onsets are marked by bold crosses. Multiplication of the corresponding frame indices with the hopsize r yields a set of strictly monotonically increasing onset times $B = \{b_1, b_2, \dots, b_N\}$ for onset-based swing ratio estimation.

3.2 Onset-Based Swing Ratio Estimation

Once we obtained a sequence B of RC onsets, we estimate s_r in a tempo-informed manner. Assuming a roughly constant tempo τ_e throughout the excerpt, the time interval $\delta = \tau_e^{-1}$ between two consecutive beats should be close to $\delta_b + \delta_o$. To account for small deviations from the ideal beat period δ , we introduce a tolerance $\alpha \geq 1$. Now, we go through every previously detected RC onset and test the hypothesis that it could be the first in a series of three consecutive onsets (backbeat, offbeat, downbeat). We denote this sub-sequence as $B_n = \{b_n, b_{n+1}, b_{n+2}\}$, $B_n \subset B$ and refer to it as onset triple. From all possible triples B_n , $n \in [1 : N - 2]$ we accept the ones that fulfill the criterion

$$(b_{n+2} - b_n) < \alpha \cdot \delta \quad (4)$$

as instances of triples embedded in an RC pattern. The swing ratio is estimated from a valid onset triple by setting $\delta_b = b_{n+1} - b_n$ and $\delta_o = b_{n+2} - b_{n+1}$ in Eqn (1). In Figure 2, we illustrate this procedure. All RC onset candidates are marked by black crosses but only the triples that fulfill the constraint in Eqn (4) are marked with different colors. Above the third triple (blue note symbols) we depict the extent of the search range $\alpha \cdot \delta$ that covers both δ_b and δ_o . As indicated in the plot, we try to find multiple

occurrences of the RC pattern triples per excerpt, so we can obtain a more robust estimate for the swing ratio by averaging over the individual s_r -values computed for each triple. For that reason, we also accept variations of the RC pattern where the offbeat impulse occurs in succession to the downbeat instead of the backbeat. As will be explained in Section 4.4, there are situations where estimation of s_r from RC onsets may deliver erroneous results. To obtain more robust estimates, we introduce LLACF-based swing ratio estimation in the next two sections.

3.3 LLACF Mid-Level Representation

We propose to employ the LLACF as a tempo-normalized mid-level representation capturing the swing ratio that is implicitly encoded in the peaks of Δ . Using the LLACF, we can circumvent the selection of onset candidates and instead transform the complete Δ into a phase-invariant, tempo-normalized representation. Swing ratio estimation then boils down to matching this representation to LLACFs with known swing ratios (see Section 3.4). To this end, we first compute a normalized ACF from the novelty function Δ as:

$$R_{\Delta\Delta}(\ell) = \frac{\sum_{m=1}^{M-\ell} \Delta(m)\Delta(m-\ell)}{\sum_{m=1}^M \Delta(m)^2}, \quad (5)$$

where we only consider the positive lags $\ell \in [0 : M - 1]$. Note that $R_{\Delta\Delta}(\ell) = R_{\Delta\Delta}(-\ell)$ due to symmetry. Moreover, $R_{\Delta\Delta}(0) = 1$ and $R_{\Delta\Delta}(\ell) < 1$ for $\ell \in [1 : M - 1]$. Each lag can be expressed as tempo value by the relation $\tau = \frac{60}{r\ell}$. We now define a logarithmically spaced tempo (log-tempo) axis, that has equal distance q between tempo octaves and has the reference tempo τ_r at a defined position. After correction for the ratio between the excerpt's tempo estimate τ_e and the reference tempo τ_r , we use linear interpolation to warp $R_{\Delta\Delta}$ onto this axis, yielding our tempo-normalized LLACF \mathcal{A} . Despite using a log-tempo axis, we stick to the term log-lag ACF since the inverse relation $\ell = \frac{60}{r\tau}$ retains the logarithmic spacing, just in opposite direction.

In the bottom row of Figure 1, we show the LLACFs corresponding to the prototypical RC patterns. Variation of s_r gives an intuition how the salience of different periodicities in the RC pattern is represented by the LLACF. Since τ_r is constant, all three LLACFs have clear peaks at the beat periodicity (240 BPM) and its integer subdivisions. For $s_r = 1$ in Figure 1(a), there is a strong peak at 480 BPM (corresponding to the straight eighth notes). With increasing swing ratio, this peak diverges into two lobes that move to other periodicities. In Figure 1(c), the first peak resides at 960 BPM (offbeat equals a sixteenth note) and the second peak is at 320 BPM (backbeat equals a dotted eighth note).

3.4 LLACF-Based Swing Ratio Estimation

In order to estimate a swing ratio from the shape of \mathcal{A} , we construct a set $\mathcal{A}_{s_r}, s_r \in \mathbb{R}$ with $1 \leq s_r \leq 4$ of prototype

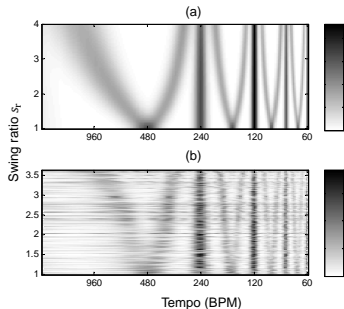


Figure 3. Evolution of the LLACF computed from RC patterns with increasing swing ratio. (a): LLACFs derived from novelty functions of idealized prototype RC patterns at a reference tempo τ_r of 240 BPM. (b): LLACFs extracted from our test corpus that have been warped to match τ_r .

LLACFs. They are extracted from novelty functions of idealized RC patterns with fixed reference tempo τ_r and varying swing ratio s_r (cf. the time-domain plots in Figure 1). In Figure 3(a) we show the complete set of prototype LLACFs with the log-tempo axis in BPM and the swing ratio increasing from bottom to top. Darker shade of gray corresponds to higher periodicity salience. One can clearly see how the offbeat-related peaks change their periodicity with the swing ratio while the peaks related to the beat (and subdivisions thereof) reside at the same periodicity.

Now, our approach to swing ratio estimation is to compare the extracted \mathcal{A} to each of these prototype LLACFs and to select the swing ratio corresponding to the best match. For the comparison, we employ Pearson's correlation coefficient. We have to take into account that the tempo estimate τ_e used for warping the LLACF to the reference log-tempo axis underlying \mathcal{A}_{s_r} may be slightly inaccurate. As a consequence, the resulting \mathcal{A} might exhibit a constant offset with respect to the prototype \mathcal{A}_{s_r} . Thus, we shift the \mathcal{A} against the log-tempo axis of each \mathcal{A}_{s_r} in a restricted interval $[-q \cdot \log_2(\alpha) : +q \cdot \log_2(\alpha)]$ to find the best alignment. Finally, the s_r corresponding the maximum correlation coefficient over all entries in \mathcal{A}_{s_r} is selected.

4 Evaluation

In this section, we describe the setup, metrics, and results of the experiments we conducted in order to compare manual, onset-based, and LLACF-based swing ratio estimation. In addition, some trends visible in the data are discussed.

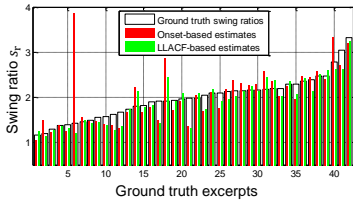


Figure 4. Comparison of the swing ratios estimated from ground truth RC onsets, automatically detected RC onsets and LLACF analysis.

4.1 The Weimar Jazz Database

The Weimar Jazz Database¹ consists of 299 (as in July 2015) transcriptions of instrumental solos in jazz recordings performed by a wide range of renowned jazz musicians. The solos have been manually annotated by musicology and jazz students at Liszt School of Music Weimar as part of the Jazzomat Research Project.² Several music properties are annotated, most notably the pitch, onset and offset of all tones played by the soloists, as well as a manually tapped beat grid, chords, form parts, phrase boundaries, and articulation. For our work, we only use the beat grid. From the complete Weimar Jazz Database, we automatically selected a subset of 921 excerpts that had been labeled with swing feel. Because we will compare the swing ratios of drummers and soloists in our future work, the excerpts had to contain at least 5 consecutive eighth notes played by the soloists. The total playtime of the selected excerpts amounts to roughly 50 minutes (out of 8 hours), their average duration is 3.3 seconds.

4.2 Evaluation Setting

A subset of 42 excerpts have been manually annotated for RC onsets in order to create a ground truth for swing ratio estimation. The reference onsets were transcribed by two experienced student assistants of the Jazzomat Research Project using the software Sonic Visualiser [5]. The ground truth subset was split in two, approximately equal parts and each part was given to one of the annotators. In total, 834 RC onsets were manually annotated. In our evaluation (cf. Sections 4.3, 4.4, and 4.5), we used the well-known metrics recall, precision and F-measure for quantitative evaluation. In order to count an onset candidate as true positive, we allowed a maximum deviation of ± 30 ms to the ground truth onset time. Furthermore, we used Pearson’s correlation coefficient as a means to quantify the agreement between reference swing ratios and automatically estimated swing ratios. We fixed the following extraction parameters for the automatic estimation of swing

ratios: The STFT blocksize w was appr. 46 ms and the hop-size r was appr. 5.8 ms. The compression-constant γ was 1000, the lower cutoff k_0 was set to equal appr. 12.9 kHz, the reference tempo τ_r was 240 BPM, the LLACF octave-resolution q was 36. The tolerance α for tempo deviations was 1.2.

4.3 Cross-Validation

At first, we are interested in the agreement between our human annotators, since we suspect that there may be ambiguous cases where it is not clear where an RC onset is exactly located in time or if there is an onset at all. Thus, we selected a small subset of 11 excerpts for which the annotators created a cross-validation transcription. Running these against the larger set, we receive an F-measure of appr. 0.96. The average absolute time difference between matched onsets in the reference and the cross-validation set amounts to 7.8 ms.

4.4 Onset-Based Evaluation

Next, we used the previously validated ground truth annotations as reference to assess the performance of our automated RC onset detection described in Section 3.2. In this scenario, we received an F-measure of appr. 0.93 and an average onset deviation of 2.5 ms. Since these results seem surprisingly good, we wanted to quantify how much potential onset detection errors would propagate into the swing ratio estimation. Using the procedure described in Section 3.2, we determined ground truth swing ratios for all manually annotated excerpts. When we compared these to the swing ratios estimated from automatically detected RC onsets, we yielded a correlation coefficient of appr. 0.66 (see Figure 4). With regard to the comparably high F-measure obtained for the onset detection, this unsatisfactory result may seem surprising at first, but can be explained using the example in Figure 2. There, we see that only 12 out of 18 RC onsets are considered for swing ratio estimation. Intuitively, small deviations in the detected onset times can lead to under- or overestimation of the swing ratio, especially for fast tempi, where subtle timing differences may get lost due to the coarse sampling of the analysis frames. Even worse errors may be caused by spurious onsets that fulfill the threshold criterion but are actually not RC patterns. This is the case for the sixth excerpt in Figure 4, where some sort of RC swell is mistaken for an onset triple, leading to a overestimation of s_r .

4.5 LLACF-Based Evaluation

Since we found the correlation between ground truth swing ratios and onset-based swing ratios to be unsatisfactory, we repeated the comparison with respect to swing ratios estimated from the LLACF as described in Section 3.3. This time, we received a correlation coefficient of appr. 0.9. In Figure 4, one can see that both methods behave similar

¹ <http://jazzomat.hfm-weimar.de/dbformat/dboverview.html>

² <http://jazzomat.hfm-weimar.de/>

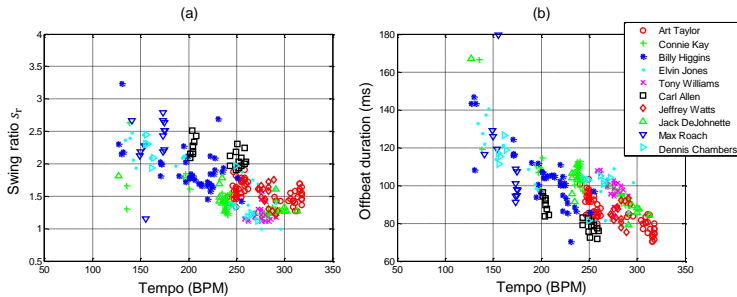


Figure 5. Scatter plots showing the relationship of tempo vs. (a): swing ratio and (b): offbeat duration. Each marker corresponds to one jazz excerpt. We only show the 10 most frequently represented drummers.

but the onset-based swing ratios exhibit some pronounced outliers. Moreover, Figure 3 shows that the prototypical LLACFs in \mathcal{A}_{s_T} correspond quite well to the LLACFs extracted from our test corpus. Both plots depict the LLACFs ordered by the corresponding swing ratio. The typical structure of periodicity peaks is clearly visible, although the LLACFs extracted from the jazz excerpts are much more noisy than the idealized LLACFs. This leads us to the conclusion that the LLACF-based swing ratio estimation is a reliable method that should be preferred over the onset-based swing ratio estimation.

4.6 Comparison to Friberg and Sundström

In Section 1, we already indicated our aim to re-examine the findings of Friberg and Sundström [9] on a larger scale. As can be seen in Figure 5(a), our automatically estimated swing ratios show similar trends as the manually annotated data used in the original paper. However, while Friberg and Sundström only had around 40 excerpts from various pieces of four drummers, we are able to study several hundreds of RC patterns played by a wide range of drummers due to our automated method (three among them—Tony Williams, Jack DeJohnette, and Jeffrey Watts—were examined by Friberg and Sundström, too).

In Figure 5, we show the results obtained for the 10 drummers represented with the most excerpts. Each point in the scatterplots is placed according to (a) s_T vs. τ_e and (b) δ_0 vs. τ_e . In general, the negative correlation of swing ratio and tempo is clearly discernable—for the whole data set as well as for certain drummers like Elvin Jones or Billy Higgins, who vary their swing ratio from appr. 2.5 around 150 BPM to appr. 1.5 at 250 BPM, and in the case of Jones even to around 1.0 at 300 BPM. However, there are also drummers who seems to keep almost the same swing ratio at different tempi, e.g., Art Taylor or Carl Allen. Additionally, Friberg and Sundström report the duration between the offbeat impulse and the next beat to be roughly constant at 100 ms for all tempi faster than 150 BPM (cf. [9, p. 337]). In general, this finding is supported by

our data (see Figure 5(b)), but the offbeat durations have a wider range from 110 ms to 80 ms and even 70 ms.

5 Conclusions and Future Work

In this paper, we presented a microtiming study conducted on a subset of the publicly available Weimar Jazz Database. Future work will be directed towards extending our method to more drummers and other recordings as well as to the comparison between RC patterns and soloists. Exact onset times of all tones of the soloists, and thus their microtiming and swing ratio, are at hand within the Weimar Jazz Database. A comparison between drummers’ and soloists’ microtiming will allow for a larger scale re-examination of one of the central findings in [9]: The swing ratio of soloists is in general lower than the swing ratio of the accompanying drummer since soloists deliberately play behind the beat while synchronizing the offbeat with the drummer. They do so, because, as Friberg and Sundström claim, “delayed downbeats and synchronized offbeats may create both the impression of the laid-back soloist, which is often strived for in jazz, and at the same time an impression of good synchronization” [9, p. 345]. Therefore, using microtiming data from the Weimar Jazz Database as well as automatically estimated swing ratios of RC patterns may lead to new insights in the interactive art of improvising together in a professional jazz ensemble.

6 Acknowledgments

The Jazzomat Research Project is supported by the German Research Foundation (Melodisch-rhythmische Gestaltung von Jazzimprovisationen. Rechnerbasierte Musikanalyse einstimmiger Jazzsoli, DFG-PF 669/7-1). The authors would like to thank all student assistants participating in the transcription and annotation process. The International Audio Laboratories Erlangen (AudioLabs) is a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer IIS.

7 References

- [1] Fernando Benadon. Slicing the beat: Jazz eighth-notes as expressive microrhythm. *Ethnomusicology*, pages 73–98, 2006.
- [2] Paul F. Berliner. *Thinking in Jazz. The Infinite Art of Improvisation*. University of Chicago Press, 1994.
- [3] Walter Gerard Busse. Toward objective measurement and evaluation of jazz piano performance via midi-based groove quantize templates. *Music Perception*, 19(3):443–461, 2002.
- [4] Matthew W. Butterfield. Why do jazz musicians swing their eighth notes? *Music Theory Spectrum*, 33(1):3–26, 2011.
- [5] Chris Cannam, Christian Landone, and Mark B. Sandler. Sonic visualiser: An open source application for viewing, analysing, and annotating music audio files. In *Proc. of the International Conference on Multimedia*, pages 1467–1468, Florence, Italy, 2010.
- [6] Geoffrey L. Collier and James Lincoln Collier. A study of timing in two louis armstrong solos. *Music Perception*, 19(3):463–483, 2002.
- [7] Mark C. Ellis. An analysis of ‘swing’ subdivision and asynchronization in three jazz saxophonists. *Perceptual and Motor Skills*, 73(3):707–713, 1991.
- [8] Arndt Eppler, Andreas Männchen, Jakob Abeßer, Christof Weiß, and Klaus Frieler. Automatic style classification of jazz records with respect to rhythm, tempo, and tonality. In *Proc. of the Conference on Interdisciplinary Musicology (CIM)*, December 2014.
- [9] Anders Friberg and Andreas Sundström. Swing ratios and ensemble timing in jazz performance: Evidence for a common rhythmic pattern. *Music Perception*, 19(3):333–349, 2002.
- [10] Peter Grosche and Meinard Müller. Extracting predominant local pulse information from music recordings. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(6):1688–1701, 2011.
- [11] Peter Grosche, Meinard Müller, and Frank Kurth. Cyclic tempogram – a mid-level tempo representation for music signals. In *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 5522–5525, Dallas, Texas, USA, March 2010.
- [12] Matthias Gruhne and Christian Dittmar. Improving Rhythmic Pattern Features Based on Logarithmic Pre-processing. In *Proc. of the Audio Engineering Society Convention (AES)*, Munich, Germany, May 2009. Preprint 7817.
- [13] André Holzapfel and Yannis Stylianou. A scale transform based method for rhythmic similarity of music. In *Proc. of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 317–320, April 2009.
- [14] André Holzapfel and Yannis Stylianou. Scale transform in rhythmic similarity of music. *IEEE Transactions on Audio, Speech & Language Processing*, 19(1):176–185, 2011.
- [15] Henkjan Honing and W. Bas de Haas. Swing once more: Relating timing and tempo in expert jazz drumming. *Music Perception: An Interdisciplinary Journal*, 25(5):471–476, 2008.
- [16] Jesper Højvang Jensen, Mads Græsbøll Christensen, and Søren Holdt Jensen. A tempo-insensitive representation of rhythmic patterns. In *Proc. of the European Signal Processing Conference (EUSIPCO)*, pages 1509–1512. Glasgow, Scotland, August 2009.
- [17] Franz Kerschbaumer. *Miles Davis: Stilkritische Untersuchungen zur musikalischen Entwicklung seines Personalstils*. Studies in jazz research. Akademische Druck und Verlagsanstalt, 1978.
- [18] Ugo Marchand and Geoffroy Peeters. The modulation scale spectrum and its application to rhythm-content description. In *Proc. of the International Conference on Digital Audio Effects (DAFx)*, pages 167–172, Erlangen, Germany, September 2014.
- [19] Will Parsons and Ernest Cholakakis. It dont mean a thing if it aint dang, dang-a dang! *Downbeat*, 52(8):61, 1995.
- [20] Geoffroy Peeters. Rhythm classification using spectral rhythm patterns. In *Proc. of the International Conference on Music Information Retrieval (ISMIR)*, pages 644–647, London, UK, September 2005.
- [21] Martin Pfeleider. *Rhythmus: Psychologische, theoretische und stilanalytische Aspekte populärer Musik*. Transcript, 2006.
- [22] Peter Reinholdsson. Approaching jazz performances empirically. some reflections on methods and problems. *Action and perception in rhythm and music*, 55:105–125, 1987.
- [23] Richard Franklin Rose. *An Analysis of Timing in Jazz Rhythm Section Performances*. PhD thesis, University of Texas, 1989.
- [24] Thomas Völkel, Jakob Abeßer, Christian Dittmar, and Holger Großmann. Automatic genre classification on latin music using characteristic rhythmic patterns. In *Proc. of the Audio Mostly: A Conference on Interaction with Sound*, pages 16:1–16:7, Piteå, Sweden, September 2010.

Poster Session 2

MUSICAL OFFSET DETECTION OF PITCHED INSTRUMENTS: THE CASE OF VIOLIN

Che-Yuan Liang, Li Su, Yi-Hsuan Yang

Academia Sinica

{mister2dot4, lisu, yang}@citi.sinica.edu.tw

Hsin-Ming Lin

University of California, San Diego

hs1040@ucsd.edu

ABSTRACT

Musical offset detection is an integral part of a music signal processing system that requires complete characterization of note events. However, unlike onset detection, offset detection has seldom been the subject of an in-depth study in the music information retrieval community, possibly because of the ambiguity involved in the determination of offset times in music. This paper presents a preliminary study aiming at discussing ways to annotate and to evaluate offset times for pitched non-percussive instruments. Moreover, we conduct a case study of offset detection in violin recordings by evaluating a number of energy, spectral flux, and pitch based methods using a new dataset covering 6 different violin playing techniques. The new dataset, which is going to be shared with the research community, consists of 63 violin recordings that are thoroughly annotated based on perceptual loudness and note transition. The offset detection methods, which are adapted from well-known methods for onset detection, are evaluated using an onset-aware method we propose for this task. Result shows that the accuracy of offset detection is highly dependent on the playing techniques involved. Moreover, pitch-based methods can better get rid of the soft-decaying behavior of offsets and achieve the best result among others.

1. INTRODUCTION

In the literature, offset detection has been frequently mentioned in the context of performance analysis [14], automatic music transcription (AMT) [4, 13, 21, 24, 29], note segmentation [10, 15, 18, 26], and computational auditory scene analysis (CASA) [19]. In these systems, offset detection is required for complete measurements of duration, intonation, vibrato, dynamics, and other kinds of note-based properties of music [14]. However, to date, offset detection is mostly treated as a component in a large system. Few studies, if any, are dedicated to offset detection.

The challenges of offset detection can be illustrated by the attack-decay-sustain-release (ADSR) model of music signals. First, consider the ADSR envelope of a plucked

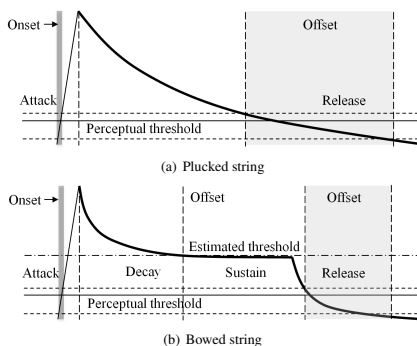


Figure 1. The ADSR envelopes of a plucked string (upper) and a bowed string signal (lower). The gray blocks show the ambiguity of onset (dark) and offset (light) due to the variation of hearing threshold. The bold-line segments of the envelopes are the possible regions to detect an offset.

string signal in Figure 1(a). The envelope of such signals usually consists of a short attack, unobservable sustain, and a gradual decay right before the release. Due to the difference in hearing threshold among human listeners, the possible region of perceptual offset time (*i.e.* medium gray region) can be fairly wide due to the gentle slope of the release. Because of this, offset detection may slip into the game of comparing the subjective listening thresholds. In contrast, there is little ambiguity associated with the onset time (*i.e.* dark gray region) due to the short attack.

Figure 1(b), on the other hand, shows the possible ADSR envelope of a bowed string signal, which contains four discernible parts. Because the release time is shorter, the temporal uncertainty of the perceptual threshold of such signals should be less than that of plucked string signals. In practice, however, computationally estimating the perceptual threshold in bowed string signals may not be easy, due to the similar shapes of the decay and the release parts. Things are more complicated in real-world signals that contain rich variation in the employed instruments and playing techniques, which would shape the ADSR envelope in totally different ways. Indeed, the challenges of offset detection can be attributed to the gentle slope of the release part and the rich variation in timbre in music signals.

© Che-Yuan Liang, Li Su, Yi-Hsuan Yang, Hsin-Ming Lin. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Che-Yuan Liang, Li Su, Yi-Hsuan Yang, Hsin-Ming Lin. "Musical Offset Detection of Pitched Instruments: The Case of Violin", 16th International Society for Music Information Retrieval Conference, 2015.

This paper presents a preliminary attempt focusing on musical offset detection. Specifically, this paper discusses various aspects of offset detection research, from building a dataset, designing an algorithm informed by the aforementioned challenges, to evaluating the performance of offset detection. We restrict our discussion on the violin, and investigate the offset detection of its six different playing techniques. This way, we exclude musical signals with very long releases, such as the pedaled piano.

Specifically, we discuss possible approaches to manually annotating offset times in music and then propose a new one (see Section 3). The proposed approach is adopted to construct a new offset detection dataset, which we have made available to the research community online.¹ With the new dataset, we present and evaluate a number of offset detection algorithms based on the spectral flux, energy and pitch attributes of music (see Section 4). To investigate the effect of playing techniques, an in-depth technique-by-technique discussion is also presented (see Section 5). As another contribution of this paper, a new evaluation measure for offset detection is also proposed and discussed.

2. RELATED WORK

Most of the offset detection algorithms are implemented in two main directions: thresholding on energy saliency, and thresholding on pitch saliency. The energy saliency can be the physical, perceptual or pitch-wise sound levels [15, 16, 20]. The thresholding on pitch saliency is often seen in the context of AMT [13, 24], where the offset can be regarded as the falling position of a pitch saliency function for a specific pitch. In Non-negative matrix factorization (NMF)-based AMT, the offset is usually determined by a threshold on the activation matrix [29]. Other approaches, such as novel features like coreentropy [10], data-driven models such as the hidden Markov models (HMM) [4, 14], support vector machines (SVM) [18], have also been applied to offset detection. Spectral flux-based approaches (*i.e.* using temporal difference of spectrum-based representations) [3, 6, 7, 28], despite being a conventional method in onset detection, are rarely used in offset detection except for some studies [19, 21, 26]. Post-processing with known onset information is sometimes used [10, 15].

3. DATASET CONSTRUCTION

3.1 Annotating the Offset

There are several possible ways to annotate the offset of musical notes and build a dataset, depending on the data format of the music content. For example, one can take the timestamps of note-off message in MIDI as the ground truth for offset. Although audio data for experiments can be generated by MIDI efficiently, this method cannot accurately indicate the perceptual offset time in many cases. For example, a control message “sustain pedal” makes the synthesizer prolong the amplitude envelope even after the

note-off message. In this case, the perceptual offset can fall far behind the note-off message. Alternatively, one can also construct a music dataset from video recordings. Video can plausibly provide visual clues to a performer’s movement which are sometimes helpful to estimate the offset time. Inaccuracy, however, may result from audio-visual asynchrony and low frame rates.

Another useful way to specify the offset times is to annotate on the spectrogram of waveform with the aid of audio visualization and musical signal analysis tools such as Sonic Visualiser. This method, however, may not be reliable due to the mismatch between the physical and perceptual offset. For example, human has varied audibility threshold in different pitch frequency ranges. Perceptual limitations, such as simultaneous masking and temporal masking, may also affect. Therefore, a more practical way is to incorporate visualization software and the hearing perception of musicians, despite the cost may be higher. Since their is no procedure for such a perception-based offset annotation, we propose a new one below.

3.2 Proposed Offset Annotation Procedure

Considering the perceptual aspects of pitched instruments, the validity of our annotation is based on two assumptions: First, if a note onset and its fundamental frequency (F0) are both retrieved, its offset time is the first moment when the sound intensity level is below the auditory threshold for a certain period of time. Second, for continuous notes, the sound intensity level may always be above the threshold. Therefore, the offset time of preceding note should be exactly or very close to the onset time of the subsequent note unless there are polyphonic notes.

With the aid of a visualization tool such as the Audacity, we propose the following steps for annotating offset times.

1. Remove DC offset (bias) and normalize maximum amplitude to -1.0 dB (software default value). This is done by the “normalize” function in Audacity.
2. Transcribe all identifiable pitches, excluding unstable overtones and unidentifiable sound resulting from playing faults or specific playing techniques (e.g. *flageolet* or *sul ponticello*).
3. Carefully and repeatedly listen to a short part of sound sample as well as zoom in the display of waveform in order to catch the onset position.
4. Identify the position within a pitch where we find the start of “attack” of amplitude envelope in the waveform. The timestamp corresponds to the note onset.
5. Catch the first perceived disappearance (*i.e.* below the audibility threshold) of that given pitch. The corresponding timestamp is the note offset time.
6. For continuous notes, we simply find consequent note onset time and use it as the preceding note offset time. However, in case of a clear note overlapping, we annotate the onset and the offset independently.
7. If the time is still not assured, we play the sound at slower speeds and repeats steps 3–6. This is helpful in estimating note onset or offset precisely.

¹ http://mac.citi.sinica.edu.tw/offset_detection/

Technique	# of clips	# of offsets
<i>Pizzicato</i>	13	144
<i>Spiccato</i>	5	168
<i>Sordino</i>	10	539
<i>Flageolet</i>	8	48
<i>Sul tasto</i>	12	140
<i>Sul ponticello</i>	15	187
Total	63	1,226

Table 1. Detailed information of the proposed dataset.

3.3 Proposed Dataset

The dataset contains 63 violin solo excerpts with a total of 1,226 notes derived from the YouTube video clips in [27] and several sound clips from the website “CompositionToday.com” [1]. This dataset, however, does not include information about music score, fingering, dynamics, vibrato, recording environment acoustics, etc. The excerpts covers 6 playing techniques, namely *flageolet* (harmonic), *pizzicato* (pluck the string), *sordino* (mute), *spiccato* (bounce the bow), *sul ponticello* (bow nearing the bridge) and *sul tasto* (bow nearing the fingerboard), all of which are widely used in orchestration [2]. These techniques produce various patterns of temporal envelopes, thereby providing a practical reference set for evaluating offset detection algorithms. Detailed information about the number of clips and notes for each playing techniques is listed in Table 1. We consider these techniques because the dataset is intended to be used as an extension of our previous work [27]. For more comprehensive experiments, people need to include more playing techniques such as legato and detache.

We hired a professional musician to annotate the dataset. The musician has profession-level training in music school and has more than 20 years of experience in playing musical instruments. He also has long experience in composing string quartet and orchestral work, and in sound mixing and recording technology. From the musician’s feedback, finishing a precise note annotation and double-check costs 1 to 2 minutes through the above process.

4. METHOD

In our study, features are extracted from three different aspects of music, including fundamental frequency, energy envelope and magnitude spectrum. We evaluate the three aspects separately to investigate their feasibility for offset detection. Revising a few previous approaches for onset detection based on these aspects, we discuss five possible offset detection algorithms in the following subsections.

4.1 Fundamental frequency

In what follows, we denote f_{0n} as the fundamental frequency at the frame index n . The corresponding MIDI number m_n can be obtained by the relation $m_n = \lfloor 12 \cdot \log_2(f_{0n}/440) \rfloor + 69$.

We adopt the spectral-domain YIN algorithm [9] to estimate the fundamental frequency. The algorithm reduces

the computation complexity of the original, time-domain YIN algorithm [12], and can produce efficient and robust estimate of fundamental frequency. It estimates the fundamental frequency by finding the minimum of the tapered square difference function $d_n(\tau)$ below a certain threshold. The function $d_n(\tau)$ is formulated as:

$$d_n(\tau) = \frac{2}{N} \sum_{k=0}^{N/2+1} |(1 - e^{2j\pi k\tau/N})\mathbf{X}_n(k)|^2, \quad (1)$$

where τ is the time lag, and $\mathbf{X}_n(k)$ is the short-time Fourier transform (STFT) spectrum at frame index n . The window size of STFT is set to $N = 2048$ in our implementation.

The minimum of Eq.(1) indicates the periodicity. The smaller the $d_n(\tau)$ is, the higher the confidence that the input signal has a fundamental frequency at $1/\tau$. Conversely, if $d_n(\tau)$ is too high then the input signal is considered non-pitched. The fundamental frequency f_0 is represented as:

$$f_{0n} = \left(\arg \min_{\tau} d_n(\tau) \right)^{-1} \quad \text{s.t.} \quad 1 - d_n(\tau) > \delta_c. \quad (2)$$

We consider the term $c_n = 1 - \min d_n(\tau)$ as the *pitch confidence*; as it measures whether an input is periodic and therefore can determine whether it is a pitch signal [9, 25]. In our implementation, we set the pitch to zero (*i.e.* $m_n = 0$) if the confidence is below a threshold δ_c . We set $\delta_c = 0.7$ empirically.

4.1.1 Pitch change

Pitch change has been known as a useful onset detector for pitched non-percussive instruments like bowed strings, where the input signal is usually excited constantly and exhibits no obvious amplitude or phase variation [11, 17]. Pitch change is a clear indicator of a note transition, which typically contains an offset of the previous note and the onset of the latter note. When the pitch contour changes from one pitch to another, we expect that there should be one note ending and another note starting. We note that the limitation of this idea is that it cannot deal with the case of repeating notes.

Based on the above observation, we propose the following offset detection method using pitch change information. We consider there is an offset event at frame n , if the following two rules are satisfied:

$$\text{mod}_{12}(m_n - m_{n-1}) \geq 1 \quad \wedge \quad c_n - c_{n-1} < 0. \quad (3)$$

Similar to the onset detector proposed in [17], the modulo operator in the first rule is applied to prevent octave errors, although it also hinders the detection of transitions of octave(s). Because $m_n = 0$ when $c_n \leq \delta_c$, the first rule also captures voice/unvoice transition. We also observe that the falling moment of confidence function can indicate the chance of a stable pitch fading that enables us to distinguish offset from onset.

4.1.2 Pitch confidence

Another perspective is to directly use the pitch confidence function as an offset detector. In this case, errors of pitch

detection would not influence the performance. The basic idea is that the time instant when the pitch confidence changes from pitched to non-pitched is considered as the offset time. Therefore, this method searches for the moment that the pitch confidence falls below the threshold. In other words, there is an offset event at n , if the following conditions meet:

$$c_{n-1} > \delta_c \quad \wedge \quad c_n < \delta_c. \quad (4)$$

Please note that this method is conceptually similar to the way many NMF-based automatic transcription algorithms detect offsets: they usually detect offsets by thresholding on the activation matrix [29]. While our method uses c_n to measure pitch confidence, NMF-based methods use the value of activation to measure pitch confidence.

4.2 Energy envelope

The energy envelope as used by the human auditory system [6] has been proven to be a robust feature in many onset detection tasks [7, 8, 22]. Here we compute the energy-like temporal envelope based on this feature. The pre-processing step starts from raw STFT spectra with frame size 2048, then map into 141 sub-bands by a set of triangular filter bank equally spaced in log-scale ranging from 30 Hz to 17000 Hz. Then, the feature is scaled by the logarithm $x \mapsto \log(1+x)$. Finally, the energy-like envelope is formulated as: $E_n = \sum_k |\bar{X}_n(k)|^2$, where $\bar{X}_n(k)$ is the pre-processed spectra magnitude of bin k . Since the perceptual offset is a subjective threshold lies between the decaying phase of energy envelope, and in most cases note offset is interrupted by succeeding onset, that make the setting an absolute thresholding infeasible. Therefore, we employ the relative threshold peak-picking algorithm [5] to find the valley of energy envelope as offset.

4.3 Spectral flux

Spectral flux is one of the most common, easy-to-implement yet powerful methods for onset detection [3, 7]. It can be formulated as: $SF_n = \sum_k H(|\bar{X}_n(k)| - |\bar{X}_{n-1}(k)|)$, where $H(x) = \frac{|x|+x}{2}$ is the rectifier function, and the pre-processed spectral bins $\bar{X}_n(k)$ are the ones that are described in Section 4.2.

We are interested in whether the idea of spectral flux can be adopted for offset detection. Two *reversed* variants of spectral flux are considered:

- **Reverse rectification (SF_{rr}):** The rectifier function H in onset detection selects only the positive flux while suppresses the negative flux. Conversely, for offset detection, H is replaced by $H' = \frac{|x|-x}{2}$, which suppress all the positive flux.
- **Reverse coding (SF_{rc}):** The other setting is to compute spectral flux in the opposite direction, *i.e.*, from the future to the past: $\sum_k H(|\bar{X}_n(k)| - |\bar{X}_{n+3}(k)|)$, to reverse the raw audio signal, and apply the normal spectral flux method to the reversed signal as looking for onset in the opposite direction.

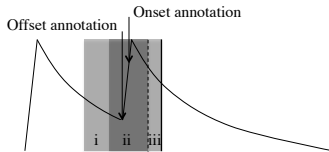


Figure 2. A case when an offset and its succeeding onset are very close, the right margin of the tolerance window for offset (the solid line) might fall behind the right margin of the tolerance window for onset (the dash line) of onset. Regions i–iii are all within the tolerance window for offset, while region ii is within the tolerance window for onset.

5. EVALUATION

5.1 Onset-aware evaluation metric

We employ the standard measures for evaluation: precision, recall and F-score. In evaluation, the offset estimate that falls within a tolerance window of length $2\delta_{\text{tolerance}}$ of the groundtruth offset time is considered to be a true positive. Moreover, the estimate and the groundtruth can only be matched at most once, based on maximum cardinality bipartite matching [23]. The remaining estimates are considered false positives. The tolerance window (centered at the groundtruth annotation) can be written as $\Delta W = [-\delta_{\text{tolerance}}, +\delta_{\text{tolerance}}]$. This is referred to as the *conventional* tolerance window.

A typical problem of this evaluation method is depicted in Fig. 2. As mentioned in Section 1, the tolerance window for offset detection is often set to be wider than that for onset detection in most previous work.² In Fig. 2, the right margin of the tolerance window for offset of the current note (*i.e.* the solid line in Fig. 2) falls behind the right margin of the tolerance window for onset of the succeeding note (*i.e.* the dash line in Fig. 2). Such a situation occurs for more than 80% of notes in our dataset, when $\delta_{\text{tolerance}}$ is set to 100ms. If the offset is annotated given the transition offset annotation rule that we suggest, region iii should not be considered as a possible true positive area.

In light of this observation, we further define a new tolerance window by $\Delta W' = [-\delta_{\text{tolerance}}, +\delta_{\text{post,tolerance}}]$, making $\delta_{\text{post,tolerance}}$ dependent on the succeeding onset. In this paper, we set $\delta_{\text{post,tolerance}} = \min(\delta_i + 50\text{ms}, \delta_{\text{tolerance}})$, where δ_i denotes the timestamp of the next onset, and 50ms is a commonly adopted value for $\delta_{\text{tolerance}}$ for onset.

To give a deep insight of the onset-aware tolerance window, let's first consider this: if the offset and succeeding onset are located far apart, the post tolerance would be the same as the conventional tolerance, so the evaluation result will be the same as the result of the conventional metric. But, as the distance becomes closer, post tolerance will shrink to the tolerance of succeeding onset when they are fully overlapped, resulting in a shortened tolerance win-

² http://www.music-ir.org/mirex/wiki/2014:Multiple_Fundamental_Frequency_Estimation_\%26_Tracking

Playing technique	Performance measure	Pitch confidence		Pitch change		Energy		SF _{rc}		SF _{rr}	
		M _A	M _B	M _A	M _B	M _A	M _B	M _A	M _B	M _A	M _B
<i>Pizzicato</i>	F-score	0.689	0.671	0.578	0.557	0.695	0.695	0.576	0.556	0.587	0.567
<i>Spiccato</i>		0.778	0.724	0.740	0.687	0.759	0.759	0.610	0.308	0.584	0.271
<i>Sordino</i>		0.727	0.718	0.701	0.686	0.555	0.512	0.650	0.598	0.652	0.596
<i>Flageolet</i>		0.381	0.381	0.321	0.301	0.414	0.402	0.292	0.262	0.290	0.254
<i>Sul tasto</i>		0.531	0.522	0.544	0.524	0.463	0.433	0.448	0.433	0.440	0.424
<i>Sul ponticello</i>		0.522	0.518	0.44	0.434	0.338	0.309	0.314	0.302	0.310	0.299
Overall		Precision	0.688	0.673	0.514	0.498	0.422	0.398	0.364	0.326	0.361
	Recall	0.623	0.609	0.669	0.648	0.639	0.604	0.758	0.677	0.759	0.674
	F-score	0.654	0.640	0.582	0.563	0.508	0.480	0.492	0.440	0.489	0.434

Table 2. Comparison of evaluation metrics to offset detection methods. M_A: the conventional evaluation metric. M_B: the proposed onset-aware evaluation metric.

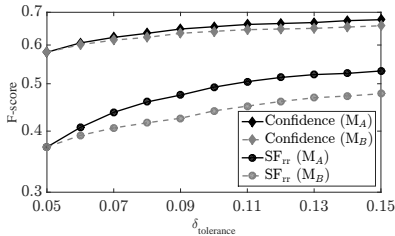


Figure 3. Comparison of evaluation using conventional metric (solid line) and proposed onset-aware metric (dashed line) on two offset detection methods. Horizontal axis shows the tolerance $\delta_{\text{tolerance}}$ (ranging from 50ms to 150ms), and vertical axis shows average F-score.

dow without region iii. In other words, the right margin of the tolerable window for offset would not exceed the right margin of the tolerable window for the succeeding onset.

5.2 Experiment result

Table 2 shows the evaluation of detection algorithms performed by both metrics. First of all, we see that pitch-based methods significantly outperform the others in the overall result according to both metrics. This is perhaps not surprising, given that pitch-based methods have been shown effective for onset detection for notes with slow attack phase. For example, Holzapfel et al. [17] have shown that pitch-based methods work much better than SF-based methods for onset detection for bow-string instrument and wind instrument. The decaying phase exhibits similar signal characteristics as soft onsets when “looking reversely” from the end of the signal. This may explain why pitch-based methods also work better than SF-based methods for offset detection.

We expect that the result of onset-aware evaluation (using $\Delta W'$) would be equal or less than conventional metric (using ΔW). The interesting finding is that, while most methods we considered have similar result for the two evaluation metrics, the result of SF-based methods degrades a lot when the onset-aware metric is adopted. For the overall result, the result of the two SF methods decreases by 11% and 13%, respectively. The most severe degradation is seen

in *spiccato*. This result indicates that SF-based methods may be prone to produce many estimations within region iii of Fig. 2.

Fig. 3 compares the result of the pitch confidence method and the SF_{rr} methods using the two metrics. As it will be shown later in Section 5.3, spectral flux exhibits temporal alignment issues while the pitch confidence method does not. It can be seen that the pitch confidence method does not suffer from the penalty of proposed metric while SF_{rr} does. We note that the bipartite matching mechanism we adopted may have also avoided some of the estimation inside region iii of Fig. 2. But, by using the proposed metric, we can ensure region iii is fully eliminated. This is important because the conventional metric may give us over-optimistic result.

Another important finding is that the pitch confidence method consistently outperforms the pitch change method, when the onset-aware metric is adopted. Results show that the pitch change method has higher recall but much lower precision, possibly due to the fluctuation of confidence above and below threshold causes some false alarms. It is possible to mitigate the issue by proper post-processing, such as by padding the continuous note or using median filter, but if the pitch confidence method is employed we do not have to deal with such an issue.

5.3 Illustration

The upper part of Fig. 4 shows the spectrogram and the offset detection functions of *pizzicato* and *spiccato*.³ Though both techniques produce sound by pulse-like excitation, we can see the envelope of *spiccato* is much smoother than *pizzicato* in terms of attack and decay phase possibly, because of the elasticity of bow cause the striking contacts the string slightly longer (*i.e.* leading to longer sustain) than the plucking string. SF based methods typically take the beginning of decay as the offset position, as shown in Fig. 2, while the estimates of other methods appear to be closer to the ground truth. SF-based methods are prone to produce temporal detection errors largely in *pizzicato* and *spiccato*, making the conventional evaluation metric for onset less appropriate for evaluate the result for offset. However, some estimations of *pizzicato* is a lot earlier than

³ We only put one of the spectral flux based methods due to their high similarity of detection curve.

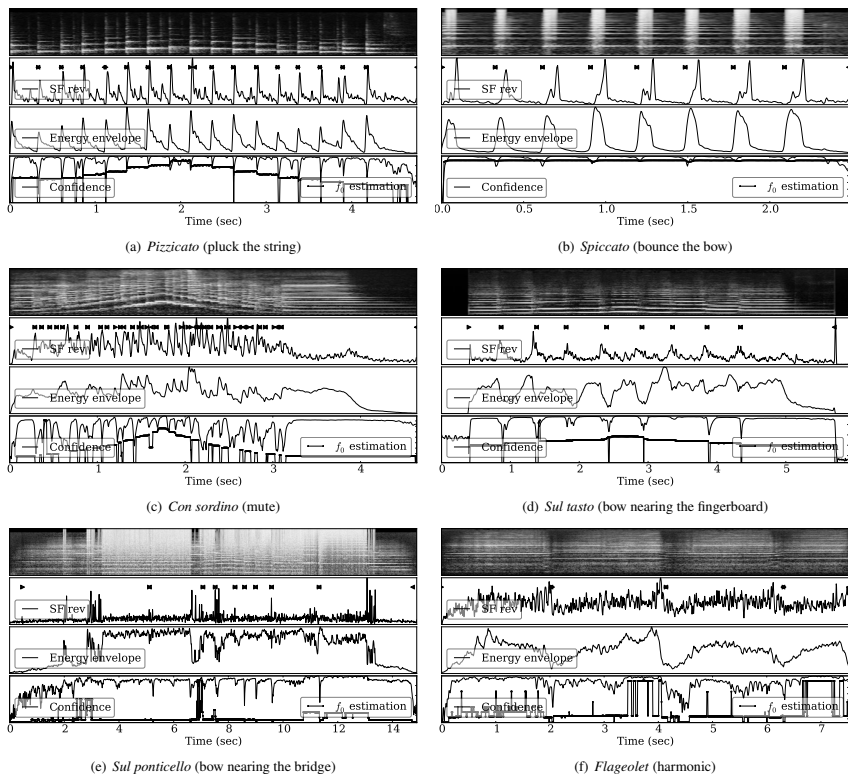


Figure 4. Comparison of the signal characteristic of six playing techniques. From top to bottom are spectrogram, spectral flux, energy envelope, and pitch-based offset detection curves. The right-pointing triangle denotes the onset annotation and the left-pointing triangle denotes the offset annotation.

spiccato that it is even located within the onset tolerance (*i.e.* region ii). In that extreme situation, we may have to shorten the onset tolerance by a few milliseconds in our evaluation metric.

The low part of Fig. 4 shows the other four bowing techniques. For the lower two techniques, all of the detection functions exhibit the fluctuating curve due to the noise-like overtones, leading to inferior result for *sul ponticello* and *flageolet*. In such case, energy relatively remains in the same level of performance. On the other hand, from the middle part of Fig. 4, we can see the pitch confidence is still a good indicator of offset for *con sordino* and *sul tasto*.

6. CONCLUSION

In this paper, we have discussed the challenges of offset detection, the methodology of constructing an offset detection dataset, some detection algorithms, and a few considerations in evaluation. Based on the newly constructed

violin dataset, we have firstly investigated the behaviors of musical offsets in the signals generated by various kinds of mechanism. We find that, in general, the pitch confidence based offset detection function outperforms algorithms based on energy and spectral flux. For the playing techniques having sharp envelopes such as *pizzicato* and *spiccato*, energy-based method can be competitive. We have also proposed an onset-aware evaluation metric that is more reliable than the conventional ones in avoiding over-estimation of true positives. We hope that these findings can contribute to the advance of research on automatic music transcription and melody tracking.

7. ACKNOWLEDGMENT

This work was supported by the Ministry of Science and Technology of Taiwan under the contracts MOST 102-2221-E-001-004-MY3, MOST 104-2221-E-001-029-MY3, and the Academia Sinica Career Development Program.

8. REFERENCES

- [1] Compositiontoday.com - sound bank - violin. http://www.compositiontoday.com/sound_bank/violin/.
- [2] S. Adler. *The Study of Orchestration—3rd Edition*. WW Norton, 2002.
- [3] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. B. Sandler. A tutorial on onset detection in music signals. *IEEE Trans. Speech Audio Proc.*, 13(5):1035–1047, 2005.
- [4] E. Benetos and S. Dixon. Polyphonic music transcription using note onset and offset detection. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Proc.*, pages 37–40. IEEE, 2011.
- [5] S. Böck, A. Arzt, F. Krebs, and M. Schedl. Online real-time onset detection with recurrent neural networks. In *Proc. Int. Conf. Digital Audio Effects*, pages 1–4, 2012.
- [6] S. Böck, F. Krebs, and M. Schedl. Evaluating the online capabilities of onset detection methods. In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 49–54, 2012.
- [7] S. Böck and G. Widmer. Maximum filter vibrato suppression for onset detection. In *Proc. Int. Conf. Digital Audio Effects*, 2013.
- [8] Sebastian Böck and Florian Krebs. MIREX onset detection task. In *Music Information Retrieval Evaluation eXchange*, 2012. [Online] <http://www.music-ir.org/mirex/abstracts/2012/BK2.pdf>.
- [9] P. M. Brossier. *Automatic annotation of musical audio for interactive applications*. PhD thesis, Queen Mary, University of London, 2006.
- [10] S. Chang and K. Lee. A pairwise approach to simultaneous onset/offset detection for singing voice using corentropy. In *Proc. IEEE Int. Conf. Acoust. Speech Signal Proc.*, pages 629–633. IEEE, 2014.
- [11] N. Collins. Using a pitch detector for onset detection. In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 100–106, 2005.
- [12] A. De Cheveigné and H. Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.
- [13] A. Degani, R. Leonardi, P. Migliorati, and G. Peeters. A pitch salience function derived from harmonic frequency deviations for polyphonic music analysis. In *Proc. Int. Conf. Digital Audio Effects*, 2014.
- [14] J. Devaney, M. I. Mandel, and I. Fujinaga. A study of intonation in three-part singing using the automatic music performance analysis and comparison toolkit (AMPACT). In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 511–516, 2012.
- [15] A. Friberg, E. Schoonderwaldt, and P. N. Juslin. Cuex: An algorithm for automatic extraction of expressive tone parameters in music performance from acoustic signals. *Acta acustica united with acustica*, 93(3):411–420, 2007.
- [16] J. Glover, V. Lazzarini, and J. Timoney. Real-time segmentation of the temporal evolution of musical sounds. In *Proc. Meetings on Acoustics*, volume 15. Acoustical Society of America, 2014.
- [17] A. Holzapfel, Y. Stylianou, A. C. Gedik, and B. Bozkurt. Three dimensions of pitched instrument onset detection. *IEEE Trans. Audio, Speech, Language Proc.*, 18(6):1517–1527, 2010.
- [18] L.-C. Hsu, Y.-L. Wang, Y.-J. Lin, C. D. Metcalf, and A. W.-Y. Su. Detection of motor changes in violin playing by emg signals. In *Proc. Int. Soc. Music Information Retrieval Conf.*, 2014.
- [19] G. Hu and D. Wang. Auditory segmentation based on onset and offset analysis. *IEEE Trans. Audio, Speech, Lang. Proc.*, 15(2):396–405, 2007.
- [20] C. Kehling, J. Abeßer, C. Dittmar, and G. Schuller. Automatic tablature transcription of electric guitar recordings by estimation of score- and instrument-related parameters. In *Proc. Int. Conf. Digital Audio Effects*, 2014.
- [21] A. Kobzantsev, D. Chazan, and Y. Zeevi. Automatic transcription of piano polyphonic music. In *Proc. Int. Symp. Image and Signal Processing and Analysis*, pages 414–418, 2005.
- [22] E. Marchi, G. Ferroni, F. Eyben, L. Gabrielli, S. Squartini, and B. Schuller. Multi-resolution linear prediction based features for audio onset detection with bidirectional lstm neural networks. In *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pages 2164–2168, 2014.
- [23] C. Raffel, B. McFee, E. J. Humphrey, J. Salamon, O. Nieto, D. Liang, and D. P. W. Ellis. mir_eval: A transparent implementation of common MIR metrics. In *Proc. Int. Soc. for Music Information Retrieval Conf.*, 2014.
- [24] J. Salamon and E. Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Trans. Audio, Speech, Lang. Proc.*, 20(6):1759–1770, 2012.
- [25] J. Serra, G. K. Koduri, M. Miron, and X. Serra. Assessing the tuning of sung indian classical music. In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 157–162, 2011.
- [26] R. Sridhar and T. V. Geetha. Raga identification of carnatic music for music information retrieval. *International Journal of recent trends in Engineering*, 1(1):571–574, 2009.
- [27] L. Su, H.-M. Lin, and Y.-H. Yang. Sparse modeling of magnitude and phase-derived spectra for playing technique classification. *IEEE/ACM Trans. Audio, Speech and Language Proc.*, 22(12):2122–2132, 2014. [Online] <http://mac.citi.sinica.edu.tw/violin-playing-technique/>.
- [28] L. Su and Y.-H. Yang. Power-scaled spectral flux and peak-valley group-delay methods for robust musical onset detection. In *Proc. Sound and Music Computing Conf.*, 2014.
- [29] E. Vincent, N. Bertin, and R. Badeau. Enforcing harmonicity and smoothness in bayesian non-negative matrix factorization applied to polyphonic music transcription. *IEEE Trans. Audio, Speech, Lang. Proc.*, 18(3):528–537, 2010.

SPECTER: COMBINING MUSIC INFORMATION RETRIEVAL WITH SOUND SPATIALIZATION

Bill Manaris

Computer Science Dept.
College of Charleston, USA
manarisb@cofc.edu

Seth Stoudenmier

Computer Science Dept.
College of Charleston, USA
stoudenmier@cofc.edu

ABSTRACT

Specter combines music information retrieval (MIR) with sound spatialization to provide a simple, yet versatile environment to experiment with sound spatialization for music composition and live performance. Through various interfaces and sensors, users may position sounds at arbitrary locations and trajectories in a three-dimensional plane. The system utilizes the JythonMusic environment for symbolic music processing, music information retrieval, and live audio manipulation. It also incorporates Iannix, a 3D graphical, open-source sequencer, for real-time generation, manipulation, and storing of sound trajectory scores. Finally, through Glaser, a sound manipulation instrument, Specter renders the various sounds in space. The system architecture supports different sound spatialization techniques including Ambisonics and Vector Based Amplitude Panning. Various interfaces are discussed, including a Kinect-based sensor system, a Leap-Motion-based hand-tracking interface, and a smartphone-based OSC controller. Finally, we present Migrant, a music composition, which utilizes and demonstrates Specter's ability to combine MIR techniques with sound spatialization through inexpensive, minimal hardware.

1. INTRODUCTION

Spec·ter (sp·k tər)

n.

1. a visible incorporeal spirit, esp. one of a terrifying nature; ghost; phantom; apparition.
2. some object or source of terror or dread.

Also, esp. Brit., spectre.

[1595–1605; < Latin spectrum; see spectrum] (<http://www.thefreedictionary.com>)

Sound spatialization offers the ability to composers and performers to specify how sounds are positioned in the listener's audio field. Most consumer-quality audio systems allow for stereo fields (i.e., the ability to pan sound from left to right channel), becoming a standard household item in the 1970s. Around the same time, quadrophonic systems were introduced (i.e., making use of 4



© Bill Manaris and Seth Stoudenmier.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Bill Manaris and Seth Stoudenmier. "Specter: Combining Music Information Retrieval with Sound Spatialization", 16th International Society for Music Information Retrieval Conference, 2015.



Figure 1. Specter system deployment at *Time Jitters* exhibit. Photo shows two people moving through the installation. Two of the four speakers are visible (top-left and center). Also visible is one of the two Kinect sensors used in the installation (top right), and one of the four projectors (each projecting to one of the four walls enclosing the installation).

channels), but did not meet with much commercial success, due to the industry's indecision to create a standard. Within the 1980s and 1990s, surround sound was introduced into the consumer market with Cinema 5.1 (6 channels), and 7.1 (8 channels), e.g., DTS, Dolby Digital, etc. Still, these systems have proprietary formats, they require specialized software and hardware, and are not as readily available (as stereo systems). Additionally, various research techniques for sound spatialization were developed during this time, including Ambisonics and Vector Based Amplitude Panning (VBAP), which have not yet received commercial acceptance.

We present Specter, an open-source, scalable, easy to customize and deploy sound spatialization system, developed to facilitate music composition and live performance.

Specter was initially developed in the context of *Time Jitters*, a four-projector interactive installation (see Figure 1), designed by Los-Angeles-based visual artist Jody Zellen for the Halsey Institute of Contemporary Art in Charleston, SC, USA.¹ *Time Jitters* includes two walls displaying looping video animation, and two walls with

¹ See a video of *Time Jitters*, <http://goo.gl/T1fpP1>

interactive elements. The concept is to create an immersive experience for participants, which confronts them with a bombardment of visual and sound elements. As participants enter the space, images and sounds are assigned to them. As participants move freely through the space, these images and sounds follow them. The result is an immersive, dynamic experience that unfolds in real-time as different people navigate the space. Several individuals contributed to this installation (including visual materials and overall concept design, interaction design, and sound design). In this paper, we focus on the design and implementation of Specter. Other aspects of this installation are presented elsewhere (e.g., [1, 2]).

Specter was designed to offer composers and performers a simple, expandable, and versatile environment to experiment with sound spatialization for music composition and live performance. It combines music information retrieval (MIR) with minimal hardware through the Open Sound Control (OSC) protocol to produce a low-cost, easily configurable and transportable system. Through various interfaces, users may position sounds at arbitrary locations and trajectories on a three-dimensional plane.

The system incorporates JythonMusic, an environment for symbolic music processing, music information retrieval, and live audio manipulation. It also utilizes Iannix, a graphical open-source sequencer for digital art for real-time generation, manipulation, and storing of sound trajectory scores. Finally, it uses Glaser, a sound manipulation instrument to render the various sounds in space by quickly manipulating their attributes.

The rest of the paper is organized as follows: Section 2 describes related background in sound spatialization. Section 3 defines the Specter system architecture; this includes a description of JythonMusic, the underlying music programming environment, used to implement Specter; Iannix, a graphical open-source sequencer for digital art, utilized to represent Specter trajectory scores; and Glaser, a sound rendering instrument. Section 4 presents a case study utilizing an MIR approach to generate a musical composition involving sound spatialization, which includes both static (pre-composed) and dynamic (interactive) sound trajectories, rendered with Specter. Finally, section 5 provides concluding remarks.

2. BACKGROUND

Although sound spatialization is a very promising field for developing new music and related composition techniques, it is highly underutilized (versus, say, timbre composition) because of the difficulty in exploring possibilities and performing existing compositions.

While there is much development already in timbre technologies for both analog and digital timbre, spatial computer music is being held back because composers have limited access to performance techniques and spaces with installed multi-channel systems [3].

Additionally, the software tools for spatial composition are few, compared to those for symbolic music, and for timbre compositions. Moreover, there is not a standard high-level format for representing spatial compositions or storing spatial recordings. As a result, spatial compositions may lose integrity and content as they are transferred from one technology to another, in order to be performed, given that the available performance spaces for spatial music are few, quite expensive to set up and maintain, have different architectures, and support different formats [4].

Nevertheless, sound spatialization is a rich field with many decades of research and development. Johnson et al. [5] provide a thorough overview of the early history of the field, starting with Schafer and Henry, who in 1951 performed the first pre-composed electroacoustic piece of music with dynamic spatialization at performance time. This was accomplished through a special interface controlling gain of individual speakers on a tetrahedral speaker array. Other important examples include construction of expensive performance spaces, during the 1970s and 1980s, such as the GRM Acousmonium, IMEB's Gmebaphone, and the Univ. of Birmingham's BEAST. Additionally, various spatialization algorithms have been developed for creating dynamic trajectories, and for spatial rendering for diffusion performance. They can be classified into two general categories:

- (a) room-based diffusion, which involves programming autonomous spatial trajectories and complex spatial distribution patterns, through large numbers of speakers; one popular approach is Higher Order Ambisonics, e.g., see [6], and
- (b) phantom-source positioning, which places less emphasis on amount of speakers, and focuses more on improving accuracy of sound object placement in the sound field, providing more control of sound trajectory rendering through better algorithms and data structures, and provision of interactive techniques for improved dynamic control of sound trajectories at performance time; one popular approach is Vector Based Amplitude Panning (VBAP), e.g. see [7].

Our system's high-level architecture supports both approaches.

Lopez-Lezcano [8] discusses development of open, general-purpose sound diffusion systems. He identifies several important characteristics of such systems, in order for them to be more usable than the current state-of-the-art. These characteristics include simplicity, transparency, versatility, using commodity hardware, using free software, and having a small footprint. Our system is designed with these characteristics in mind, as described in the next section.

A significant research trend in sound spatialization involves gestural control (e.g., [9-12]) at composition time

(such as through algorithms exploring dynamical systems, e.g., swarms and boids), but also at performance time (through specialized interfaces, such as data gloves, Kinect, and LeapMotion sensors, among others). Specter, through the underlying JythonMusic environment provides similar capabilities (a) through development of arbitrary algorithms to drive (or guide aspects of) music composition, and (b) through a variety of devices that can communicate with it via MIDI or OSC protocols.

3. SPECTER ARCHITECTURE

Specter incorporates three major components, JythonMusic, Iannix, and Glaser – all communicating via OSC to pass data and synchronize / coordinate their actions. The following sections describe each of the sub-systems, and how they are combined to provide an environment to experiment with sound spatialization for music composition and live performance.

Through various interfaces and sensors, users may position sounds at arbitrary locations and trajectories on a two- or three-dimensional plane.

3.1 Music Information Retrieval

MIR functionality is available to Specter through JythonMusic, an environment for music analysis, composition and performance (see <http://jythonmusic.org>).

JythonMusic provides libraries for music making, image manipulation, building graphical user interfaces (GUIs), and for connecting computers to external MIDI and OSC devices, such as digital pianos, smartphones, and tablets.

JythonMusic is an outcome of a decade-long project exploring various aspects of music information retrieval, including investigation of fractals in music, and their relationship to human aesthetics (e.g., [13]). This on-going project explores Zipf's Law (and related power laws) in music data mining, in music recommendation, and in music analysis, composition, and performance [14-16].

JythonMusic incorporates the following libraries. Primitives from each of these libraries are used in conjunction with Specter (as explained below) to create sound spatialization trajectories and related processes:

- **Music library** - provides primitives for creating music notes, phrases, parts, and scores, and for playing them live, as well as reading and writing them as MIDI or XML files.
- **Audio library** - provides primitives for loading and looping audio files, and for recording and looping live audio.
- **Zipf library** - provides primitives for extracting measurements from musical data (e.g., [13]).
- **MIDI library** - provides primitives for loading and looping MIDI files, and for connecting to external MIDI devices (e.g., pianos, guitars, synthesizers, etc.).

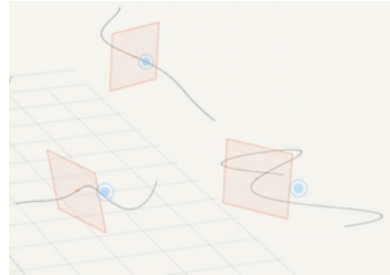


Figure 2. Example of an Iannix score consisting of three curves, each with a cursor and a trigger, drawn in 3D space. When such a score is played, cursors move automatically on prescribed trajectory, continuously reporting their XYZ coordinates, and when they cross triggers (via OSC and MIDI protocols).

- **OSC library** - provides primitives for connecting to other devices via Open Sound Control (e.g., smartphones, tablets, computers, synthesizers, etc.).

Additionally, JythonMusic provides libraries for graphical interactivity, image manipulation and sonification, and event scheduling. Finally, it encapsulates various cross-platform libraries for MIR and music / sound manipulation, such as jMusic and jSyn.

In summary, JythonMusic provides the glue code through which Specter is implemented, and, through its libraries facilitates arbitrary possibilities for data mapping, sonification, and interaction.

3.2 Sound Spatialization Trajectories

Sound spatialization trajectories in Specter are modeled via Iannix, an open-source, a 3D graphical sequencer for real-time generation, manipulation, and storing of musical and other scores (see <http://www.iannix.org>). As shown in Figure 2, Iannix scores consist of:

- **Curves**, which define spatial trajectories. These trajectories support cursors and triggers. Curves can be circular, straight lines, Bézier curves, free-form curves (drawn via mouse), or prescribed through math equations.
- **Cursors**, which follow the trajectories defined by curves, moving at a constant speed, when the score is played. Cursors report (via MIDI or OSC messages) their current coordinates in XYZ space (as defined by the Iannix score); also they can be set externally (via MIDI or OSC messages).
- **Triggers**, which report (again, via MIDI or OSC) when a cursor crosses them.

Additionally, through JythonMusic arbitrary math functions and algorithms may be implemented (such as boids, swarms, and other dynamic particle systems), which may

provide Specter with 3D trajectory information. This information may be stored as an Iannix score, or used in real-time to render sound spatialization trajectories.

3.3 Audio Rendering

Audio rendering in Specter is done through the Glaser subsystem. Glaser is an audio rendering instrument implemented using JythonMusic. It was developed for the *TimeJitters* exhibit (see section 1), and has been adapted to implement the functionality needed by Specter.

Glaser, in its basic form, allows exploring various sounds for sound design by manipulating their attributes (frequency, volume, and spatialization). It consists of three GUI displays with several sliders each, one per audio file. Through these, it allows a sound designer, composer, or performer to interactively control volume, frequency, and spatialization of an arbitrary number of audio files simultaneously.

Within Specter, the Glaser architecture has been extended to support both the Ambisonics and Vector Based Amplitude Panning approaches (e.g., see [6, 7]). This allows taking into account the spatial configuration (number of individual channels available) and geometry of the space, through existing algorithms, such as the ones used in [17].

3.4 Music Representation

Specter, through JythonMusic, utilizes a common-practice-based notation to represent audio to be rendered. This notation consists of:

- **Notes**, which specify pitch, duration, dynamic, and panning. For stereo, panning ranges from 0.0 to 1.0 (where 0.0 is left, 0.5 is center, and 1.0 is right of stereo field). Panning values greater than 1.0 are treated as identifiers for Iannix trajectories, which are used when the corresponding note is rendered). Pitches are used for sound frequency shifting (sounds are assigned a default / reference pitch, i.e., A4), durations are in seconds, and volume ranges from 0 to 127, following the MIDI standard.
- **Phrases**, which serve as containers for sequences of notes.

Additional, higher-level containers include parts and scores (again, see <http://jythonmusic.org>).

3.5 Expandable Architecture

The design concept behind Specter allows using easily accessible, low-cost equipment to render multi-channel audio, in an expandable architecture. This is facilitated by computer programming to account for the modular architecture.

Through the use of audio aggregates and low-cost, 2-channel USB audio interfaces, such as the Behringer UCB222 (approx. \$30, at the time of this writing), it is possible to assemble a wide variety of sound spatializa-

tion architectures (e.g., obviously stereo, quadrophonic, 9-channel, 16-channel, and so on). Theoretically, any number of speakers / channels is possible for arbitrary sound spatialization installations.

3.6 Interfaces for Sound Spatialization

Given the underlying functionality provided to Specter via JythonMusic, a wide variety of interfaces may be used (or developed) to generate and/or capture sound trajectories. These trajectories may be stored (in a Iannix score) for later use in audio rendering, or be used immediately for real-time sound placement (this is exemplified in the case study presented in section 4). Possible interfaces include:

- **Kinect motion sensor:** Utilizing Kinect sensors with JythonMusic has already been implemented in the context of *Kuatro*, a motion-based framework for developing interactive music installations [1].
- **LeapMotion:** We have also developed a LeapMotion interface to capture fine movement of both hands. This inexpensive sensor, and its versatile API, allow for a wide variety of interfaces and associated gestures to be developed for natural, intuitive control of sound spatialization.
- **Smartphone:** Using smartphone sensors, e.g., gyro and accelerometer readings, one may develop various programs to control aspects of musical performance. One such example is presented in the next section.

Various other possibilities exist, utilizing any type of sensor that supports MIDI and OSC protocols.

4. MIGRANT - A CASE STUDY

Migrant is a cyclic piece for piano and computer, originally composed for *Undomesticated*, a public-art installation by Vassiliki Falkehag at Moore Farms Botanical Gardens, in the context of ArtFields 2015, held in Lake City, SC, USA (<http://www.artfieldssc.org>).

It is used here as an example of combining music information retrieval techniques and sound spatialization for music composition and live performance.

Migrant is part of the ISMIR 2015 music program to be performed on Wednesday, October 28, 2015 at the Sala Unicaja de Conciertos María Cristina in Málaga, Spain.

4.1 Composition Techniques

In terms of composition, *Migrant* integrates data sonification, interactivity, and sound spatialization.

The data used in the piece comes from migrant worker statistics, including migration patterns, age, wages, family dependents, and other elements of the migrant life experience. This data was collected from 56,976 in-person interviews with hired crop farm workers. The interviews

were conducted in 545 US counties and 43 states during fiscal years 1989-2012.²

Each note in the piece represents a single person.

Melody, harmony and dynamic are all driven by the data. Data from 120 people were randomly selected. Notes were carefully spatialized – set to “fly around” to reflect the nomadic lifestyle. Different timescales were combined. A few notes were manually adjusted by the composer to reflect his own aesthetic and migrant experience. Figures 3a and b show photos used by the composer to provide aesthetic inspiration for composing the sonification scheme.

The composition makes use of the golden ratio to affect the piece’s harmonic density (e.g., see Figure 4). The sonification code was written in JythonMusic using ideas presented in [20].

A preview of the piece (one cycle, mixed for two speakers) is available here – <http://goo.gl/iYOVmY>.

The composition employs interactivity to control tempo and spatialization of notes, via a smartphone-based controller manipulated by one of the performers. This controller sends gyro readings via OSC to a JythonMusic program, similar to this – <http://goo.gl/dsTWFM>.

4.2 Performance Needs

In terms of performance, the piece requires one piano, one computer, a video projector, two (or more) speakers (as described above), and a smartphone.

The original composition envisioned eight pianos arranged in 45-degree increments, with the audience seated in the middle. For ISMIR 2015, in order to demonstrate Specter, a single piano and a computer with sound spatialization are used.

Ideally, four speakers in a square configuration (as seen in Figure 5) allow the audience to fully experience the “flying around” of notes. However, two speakers in a stereo configuration work also, albeit losing one of the sound spatialization dimensions; in this case, the outcome is similar to the preview of the piece above.

4.3 Performance Instructions

Migrant is a cyclic piece for piano and computer, using a smartphone-based OSC controller, during the performance, to send these notes “flying around” in the sound field.

Each cycle of the piece lasts 4 minutes and 52 seconds. It is meant to be played in a continuous loop - minimally two times. For the ISMIR 2015 performance, the piece will be cycled exactly twice, for a total duration of 9 minutes and 44 seconds.

During the first cycle, the pianist plays the notes in the score verbatim. The computer layers identical notes in a computer-enhanced timbre and diffuses them in a circular



Figure 3a. One of several photos used by the composer to provide aesthetic inspiration for composing the sonification scheme. *Photo Credit:* Dorothea Lange (1936), “Destitute pea pickers in California. Mother of seven children. Age thirty-two. Nipomo, California” [18].



Figure 3b. Another of the photos used by the composer to provide aesthetic inspiration for composing the sonification scheme. *Photo Credit:* Dorothea Lange (1938), “Abandoned farm with windmill and farm equipment. Dalhart, Texas. June 1938” [19].

pattern (or left-to-right pattern, for 2 speakers), thus generating a “flying-around” of notes.

The computer also displays images (using precise, scripted timings), via the video projector (again, see the piece preview provided here – <http://goo.gl/iYOVmY>).

During the second cycle, the computer plays the complete first cycle (i.e., both the notes originally played by the pianist, as well as the enhanced timbres spatialized in the sound field).

The pianist is instructed to improvise additional notes, guided by the score notes. The only constraint is that an A natural minor scale is used. No constraints are given in terms of note start times, durations, or harmony. The pi-

² See National Agricultural Workers Survey, Public Access Data, October 1, 1988 to September 30, 2012 – <http://goo.gl/xWsQe8>

anist is encouraged to create a musical narrative (to the best of their musical abilities – a challenge!), which aesthetically complements the sonified “narratives” of the people / notes in the data. In essence, this provides an opportunity for the pianist to interweave his or her own experience, aesthetic, and improvisatory skills into the piece.

During this, the computer performer utilizes the smartphone-based OSC controller to affect timing and spatialization of the computer-generated notes. This way, he or she controls aspects of the musical expression of the combined performance, through the following gestures:

- **Ready Position:** Smartphone is held facing up, parallel with the floor.
- **Controlling Tempo:** The phone is used in a percussive gesture (moving downward) to play the next note. When the phone pitch (see Figure 6) crosses the neutral (parallel to the floor) position, the next note is played.
- **Controlling Volume:** Device shake corresponds with loudness of notes. The more intensely one shakes or vibrates the phone as notes are generated, the louder the notes are.
- **Controlling Spatialization:** The yaw of the phone corresponds to placement of notes on the periphery of the sound field. (System is calibrated before the performance so magnetic north corresponds with Specter’s virtual north, as far as sound placement is concerned).

The interactive aspects of Migrant allow both human performers to musically interact with each other, and together, to interact with the musical “narrative” generated from the data.

5. CONCLUSION

Sound spatialization / diffusion systems normally require expensive, specialized equipment, which is usually hard to transport. We presented Specter, a simple, yet versatile environment to experiment with sound spatialization for music composition and live performance. By combining readily available hardware and software, through a simple, customizable architecture, we offer an inexpensive alternative to existing sound spatialization systems.

Specter may be used by MIR practitioners, as well as music composers and artists to explore and experiment with sound spatialization / diffusion more easily. Additionally, this project may facilitate development of innovative art installations, as well as new gaming experiences. Through the underlying JythonMusic system, developers may connect various MIR techniques to music composition and sound spatialization, open the door for new sonification applications, and develop innovative, immersive interactive applications (e.g., [21]).

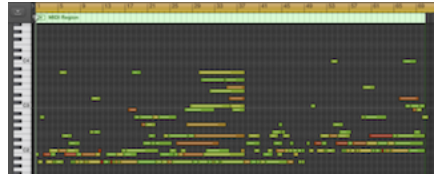


Figure 4. Pianoroll excerpt of the piece demonstrating result of sonification and use of the golden ratio to affect harmonic density.



Figure 5. Migrant performance 4-channel speaker arrangement. Audience is seated in the middle. However, a regular two-speaker (stereo) setup is possible (for convenience).



Figure 6. Smartphone Pitch, Roll, and Yaw directions.

ACKNOWLEDGEMENTS

We would like to thank Blake Stevens, Yiorgos Vassilandonakis, David Johnson, and Chris Benson for contributing to the development of some of the ideas and concepts presented herein. The first author would like to thank Vassiliki Falkehag for providing the inspiration for composing Migrant. Also, Mark Sloan and the Halsey Institute, in Charleston, SC, USA provided funding and space for the first implementation of Specter. Funding for this project has been provided in part by NSF (DUE-1323605).

REFERENCES

- [1] D. Johnson, B. Manaris, Y. Vassilandonakis, and S. Stoudenmier, "Kuatro: A Motion-Based Framework for Interactive Music Installations," *Proc. of the 40th International Computer Music Conference (ICMC 2014)*, Athens, Greece, 2014, pp. 355-362.
- [2] J. Drucker, "Pause Effect," <http://goo.gl/kLwgbY>, accessed July 16, 2015.
- [3] E. Lyon: "The Future of Spatial Computer Music," *Proc. of the 40th International Computer Music Conference (ICMC 2014)*, Athens, Greece, 2014, pp. 850-854.
- [4] M. Baalman: "Spatial Composition Techniques and Sound Spatialisation Technologies," *Organized Sound*, 15(3), pp. 209-218.
- [5] B. Johnson, M. Norris, and A. Kapur: "Diffusion Diffusion: A History of the Technological Advances in Spatial Performance," *Proc. of the 40th International Computer Music Conference (ICMC 2014)*, Athens, Greece, 2014, pp. 126-132.
- [6] Blue Ripple Sound, "HOA Technical Notes – Introduction to Higher Order Ambisonics," <http://goo.gl/srrKC6>, accessed July 16, 2015.
- [7] V. Pulkki: "Generic panning tools for MAX/MSP," *Proc. of the 26th International Computer Music Conference (ICMC 2000)*, Berlin, Germany, 2000.
- [8] F. Lopez-Lezcano: "Towards Open 3D Sound Diffusion Systems," *Proc. of the 40th International Computer Music Conference (ICMC 2014)*, Athens, Greece, 2014, pp. 869-876.
- [9] T. Davis and O. Karamanlis: "Gestural Control of Sonic Swarms: Composing with Grouped Sound Objects," *Proc. of the Sound and Music Computing Conference*, Lefkada, Greece, 2007, pp. 192-195.
- [10] M. Marshall, J. Malloch, and M.M. Wanderley: "Gesture Control of Sound Spatialization for Live Musical Performance," *Gesture-Based Human-Computer Interaction and Simulation, Lecture Notes in Computer Science*, vol. 5085, 2009, pp. 227-238.
- [11] E. Soria and R. Morales-Manzanares: "Multidimensional sound spatialization by means of chaotic dynamical systems," *Proc. of the 13th International Conference on New Interfaces for Musical Expression (NIME 2013)*, Daejeon, Korea, 2013, pp. 79-83.
- [12] B. Johnson, M. Norris, and A. Kapur: "tactile motion: An iPad Based Performance Interface For Increased Expressivity in Diffusion Performance," *Proc. of the 40th International Computer Music Conference (ICMC 2014)*, Athens, Greece, 2014, pp. 798-801.
- [13] B. Manaris, P. Roos, D. Krehbiel, T. Zalonis, and J.R. Armstrong, "Zipf's law, power laws and music aesthetics," in T. Li, M. Ogihara, G. Tzanetakis (eds.), *Music Data Mining*, pp. 169-216, CRC Press - Taylor & Francis, 2011.
- [14] B. Manaris, D. Krehbiel, P. Roos, and T. Zalonis, "Armonique: Experiments in content-based similarity retrieval using power-law melodic and timbre metrics," *Proc. of the 9th International Conference on Music Information Retrieval (ISMIR 2008)*, Philadelphia, PA, pp. 343-348, 2008.
- [15] B. Manaris, P. Roos, P. Machado, D. Krehbiel, L. Pellicoro, and J. Romero, "A Corpus-Based Hybrid Approach to Music Analysis and Composition," *Proc. of 22nd Conference on Artificial Intelligence (AAAI-07)*, Vancouver, BC, pp. 839-845, Jul. 2007.
- [16] B. Manaris, D. Hughes, and Y. Vassilandonakis, "Monterey Mirror: Combining Markov models, genetic algorithms, and power laws," *Proc. of 1st Workshop in Evolutionary Music, 2011 IEEE Congress on Evolutionary Computation (CEC 2011)*, New Orleans, LA, pp. 33-40, Jun. 2011.
- [17] R. Graham, "SEPTAR: Audio Breakout Design for Multichannel Guitar," *Proc. of the 15th International Conference on New Interfaces for Musical Expression (NIME 2015)*, Baton Rouge, Louisiana, 2015.
- [18] D. Lange, "Destitute pea pickers in California. Mother of seven children. Age thirty-two. Nipomo, California," Mar. 1936. https://en.wikipedia.org/wiki/Florence_Owens_Thompson
- [19] D. Lange, "Abandoned farm with windmill and farm equipment. Dalhart, Texas," Jun. 1938. <http://www.pbs.org/kenburns/dustbowl/photos>
- [20] B. Manaris and A. Brown, *Making Music with Computers: Creative Programming in Python*, Chapman & Hall/CRC Textbooks in Computing, 2014.
- [21] B. Manaris, D. Johnson, and M. Rourk, "Diving into Infinity: A Motion-Based, Immersive Interface for M.C. Escher's Works," *Proc. of the 21st International Symposium on Electronic Art (ISEA 2015)*, Vancouver, Canada, Aug. 2015 (to appear).

CONTENT-AWARE COLLABORATIVE MUSIC RECOMMENDATION USING PRE-TRAINED NEURAL NETWORKS

Dawen Liang, Minshu Zhan, and Daniel P. W. Ellis

LabROSA, Dept. of Electrical Engineering
Columbia University, New York

{dliang@ee., mz2468@, dpwe@ee.}columbia.edu

ABSTRACT

Although content is fundamental to our music listening preferences, the leading performance in music recommendation is achieved by collaborative-filtering-based methods which exploit the similarity patterns in user’s listening history rather than the audio content of songs. Meanwhile, collaborative filtering has the well-known “cold-start” problem, i.e., it is unable to work with new songs that no one has listened to. Efforts on incorporating content information into collaborative filtering methods have shown success in many non-musical applications, such as scientific article recommendation. Inspired by the related work, we train a neural network on semantic tagging information as a content model and use it as a prior in a collaborative filtering model. Such a system still allows the user listening data to “speak for itself”. The proposed system is evaluated on the Million Song Dataset and shows comparably better result than the collaborative filtering approaches, in addition to the favorable performance in the cold-start case.

1. INTRODUCTION

Music recommendation is an important yet difficult task in music information retrieval. A recommendation system that accurately predicts users’ listening preferences bares enormous commercial value. However, the high complexity and dimensionality of music data and the scarcity of user feedback makes it difficult to create a successful music recommendation system.

Two primary approaches exist in recommendation: collaborative filtering and content-based methods. For music, the state-of-the-art recommendation results have been achieved by collaborative filtering methods, which requires only information on users’ listening history rather than the musical content for recommendation. The central assumption of this model is that a user is likely to accept a song that is liked by users who have similar taste. A major category of collaborative filtering approaches is based on latent

factor model. It assumes that a low-dimensional representation exists for both users and songs such that the compatibility between a user and a song, modeled as their inner product in this latent space, predicts the user’s fondness of the song. In the case that user feedback is *implicit* (e.g., whether or not the user has listened to a particular song), the weighted matrix factorization from Hu *et al.* [6] works particularly well. Details regarding collaborative filtering will be further discussed in Section 2.1.

On the other hand, modeling musical content for the purpose of taste prediction is difficult due to the structural complexity present in music data which is hard to capture by simple models. Deep learning has shown its power in various pattern recognition tasks with its capability of extracting hierarchical representations from raw data. In music recommendation, van den Oord *et al.* [13] have experimented with neural networks on predicting the song latent representation from musical content.

It is natural to combine collaborative filtering and content models in recommendation to utilize different sources of information. A successful attempt from Wang and Blei [14], which joins a content model on article with collaborative filtering, achieves good performance on scientific article recommendation.

Inspired by these mentioned above, we create a content-aware collaborative music recommendation system. As the name suggests, the system has two components: the content model and the collaborative filtering model. To obtain a powerful content model, we pre-train a multi-layer neural network to predict semantic tags from vector-quantized acoustic feature. The output of the last hidden layer is treated as a high-level representation of the musical content, which is used as a prior for the song latent representation in collaborative filtering. We evaluate our system on the Million Song Dataset and show competitive performance to the state-of-the-art system.

2. RELATED WORK

In this section we review important relevant work. First we give an overview of matrix factorization model for recommendation, especially for *implicit feedback*. Then we describe two models which are closely related to ours: collaborative topic model for article recommendation and deep content-based music recommendation.



© Dawen Liang, Minshu Zhan, Daniel P. W. Ellis.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Dawen Liang, Minshu Zhan, Daniel P. W. Ellis, “Content-Aware Collaborative Music Recommendation Using Pre-trained Neural Networks”, 16th International Society for Music Information Retrieval Conference, 2015.

2.1 Recommendation by matrix factorization

A widely used approach to recommendation is collaborative filtering, where items are recommended to a user based on other users with similar patterns of item consumption. Matrix-factorization-based latent factor models [6, 8] are among the most successful collaborative filtering methods.

In a matrix factorization recommendation model, we represent both users and items in a shared low-dimensional space of dimension K , where user u is represented by a latent factor $\theta_u \in \mathbb{R}^K$ and item i is represented by a latent factor $\beta_i \in \mathbb{R}^K$. To make a prediction about the preference of user u on item i , we simply take the dot product between the two $\hat{r}_{ui} = \theta_u^T \beta_i$. To estimate user and item factors, we can minimize the squared loss between the estimated preference and actual responses $\sum_{u,i} (r_{ui} - \hat{r}_{ui})^2$, with ℓ_2 regularization on the factors to prevent overfitting. Alternating least squares (ALS) can be employed for efficient optimization. Equivalently, we can formulate a probabilistic matrix factorization model [12] with the following generative process:

- For each user u , draw user latent factor:

$$\theta_u \sim \mathcal{N}(0, \lambda_\theta^{-1} I_K),$$

- For each item i , draw item latent factor:

$$\beta_i \sim \mathcal{N}(0, \lambda_\beta^{-1} I_K),$$

- For each user-item pair (u, i) , draw feedback:

$$r_{ui} \sim \mathcal{N}(\theta_u^T \beta_i, c_{ui}^{-1}),$$

and obtain the same estimates via maximum *a posteriori*. Here c_{ui} represents our confidence on the corresponding response r_{ui} , i.e., larger value of c_{ui} indicates that there is less uncertainty about the response r_{ui} , and vice versa. This is especially crucial in the case of implicit feedback (e.g., whether user u listened to song i), because of its noisy nature. Hu *et al.* [6] propose a simple heuristic for setting the values of c_{ui} for implicit feedback¹:

$$c_{ui} = 1 + \alpha \log(1 + r_{ui}/\epsilon)$$

where α and ϵ are tunable hyperparameters. This method achieves the state-of-the-art recommendation performance in the implicit feedback case.

2.2 Collaborative topic model

Due to its content-free nature, collaborative filtering approaches can be applied in a wide range of domains. They perform well on what is called *in-matrix* predictions, i.e., recommending items that have been consumed by some users. However, this approach suffers from the well-known problem that it is unable to recommend new items that no user has consumed, or making *out-of-matrix* predictions,

¹In [6], the observational model is on the binary indicator variable $p_{ui} = \mathbb{1}\{r_{ui} > 0\}$ rather than r_{ui} , i.e., $p_{ui} \sim \mathcal{N}(\theta_u^T \beta_i, c_{ui}^{-1})$. However, in this paper the response r_{ui} is itself binary, indicating whether user u has listened to song i . Thus we treat r_{ui} and p_{ui} interchangeably.

where content-based models are better suited. Many efforts have been made to incorporate content into collaborative filtering. Wang and Blei [14] propose the collaborative topic regression (CTR) model for scientific article recommendation, which is particularly relevant to our proposed method.

There are two components in CTR: a matrix factorization collaborative filtering model (as described in Section 2.1) and a latent Dirichlet allocation (LDA) article content model. LDA [2] is a mixed-membership model on documents. Assuming there are K topics $\Phi = \{\phi_{1:K}\}$, each of which is a distribution over a fixed set of vocabulary, LDA treats each document as a mixture of these topics where the topic proportion π_i is inferred from the data. One can understand LDA as representing documents in a low-dimensional “topic” space with the topic proportion being their coordinates. With this interpretation, the generative process of CTR is as follows:

- For each user u , draw user latent factor:

$$\theta_u \sim \mathcal{N}(0, \lambda_\theta^{-1} I_K),$$

- For each document i ,

- Draw topic proportion $\pi_i \sim \text{Dirichlet}(\alpha)^2$,
- Draw latent factor $\beta_i \sim \mathcal{N}(\pi_i, \lambda_\beta^{-1} I_K)$,

- For each user-document pair (u, i) , draw feedback:

$$r_{ui} \sim \mathcal{N}(\theta_u^T \beta_i, c_{ui}^{-1}).$$

We can see CTR differs from [6] in that CTR assumes that the item latent factor β_i is close to the topic proportion π_i but could deviate from it if necessary. This allows the user-item interaction data to “speak for itself”. An attractive characteristic of CTR is its capability of making *out-of-matrix* predictions. This is done by using the topic proportion π_i alone as the item latent factor: $\hat{r}_{ui} = \theta_u^T \pi_i$, which is not possible in the traditional collaborative filtering model.

Although CTR achieves better recommendation performance than pure collaborative filtering, it does not scale well with large data. Since the model is not conditionally conjugate: the prior on β_i comes from a Dirichlet-distributed random variable π_i , topic proportion π_i cannot be updated analytically and slower numerical optimization method is required. To address this problem, Gopalan *et al.* [5] propose the collaborative topic Poisson factorization (CTPF). This model replaces the Gaussian likelihood and Gaussian prior in CTR with Poisson likelihood and gamma prior, thus becoming conditionally conjugate with closed-form updates. Experiments on large-scale scientific article recommendation demonstrate that CTPF performs significantly better than CTR.

The main difference that sets our method apart from collaborative topic model is the content model. As a feature extractor, LDA can only produce linear factors due to its bilinear nature. On the other hand, multi-layer neural network used by in our system is capable of capturing the non-linearities in the feature space.

²The generative process for words is omitted for brevity throughout the paper. Please refer to [14] for details.

2.3 Deep content-based music recommendation

Previous attempts on content-based music recommendation have achieved promising results. van den Oord *et al.* [13] utilize a neural network to map acoustic features to the song latent factors learned from the weighted matrix factorization [6]. As a result, given a new song that no one has ever listened to, a latent factor can still be predicted from the network and recommendation can be done in the same fashion as with a regular collaborative filtering model.

Our method is very similar to this approach, but we will point out two major differences:

- First, the neural network is used for different purposes. We use it as a content feature extractor, just like LDA in the collaborative topic model. The neural network in [13] maps content directly to the latent factors learned from pure collaborative filtering, and the resulting model is expected to operate similarly to collaborative filtering even when usage data is absent.
- Since the neural network is trained to map content to the latent factors learned from the weighted matrix factorization, the performance of [13] is unlikely to surpass that of the weighted matrix factorization. What we propose in this paper, on the other hand, uses content as an *addition* to the weighted matrix factorization, in a similar manner as the collaborative topic model described in Section 2.2. As we show in the experiment, we are able to achieve better result than the weighted matrix factorization when we only have limited amount of user feedback.

Other approaches that hybridize content and collaborative models include Yoshii *et al.* [17], McFee *et al.* [11], and Wang and Wang [15]. [17] train a three-way probabilistic model that joins user, item, and content by a latent “topic” variable; the model focuses on explicit feedback (user ratings). [11] take a similar approach to [13] and learn a content-based similarity function from collaborative filtering via metric learning. [15] also use a neural network to incorporate music content into the collaborative filtering model. The major difference is that in [15] the output of the neural network is treated as item factor and the neural network is trained to minimize a collaborative-filtering-based loss function. Therefore the content model itself does not have explicit musicological meaning.

3. PROPOSED APPROACH

Adopting the same structure as that of CTR, our system consists of two components: a content model which is based on a pre-trained neural network and a collaborative filtering model based on matrix factorization.

3.1 Supervised pre-training

Inspired by the success of transfer learning in computer vision which exploits deep convolutional neural networks

[9], in our system we pre-train a multi-layer neural network in a supervised semantic tagging prediction task and use it as the content model.

Our training data comes from Liang *et al.* [10] which consists of 370K tracks from the Million Song Dataset and the pre-processed *last_fm* data with a vocabulary of 561 tags, including genre, mood, instrumentation, etc. We use the Echonest’s timbre feature, which is very similar to MFCC. To get the song-level features, we vector-quantize all the timbre features following the standard procedure: We run the k -means algorithm on a subset of randomly selected training data to learn $J = 1024$ cluster centroids (codewords). Then for each song, we assign each segment (frame) to the cluster with the smallest Euclidean distance to the centroid. We aggregate the VQ feature of song i ($\mathbf{x}_i \in \mathbb{R}_+^J$) by counting the number of assignments to each cluster across the entire song and then normalize it to have unit ℓ_1 norm to account for the various lengths.

We treat music tagging as a binary classification problem: For each tag, we make independent predictions on whether the song is tagged with it or not. We fit the output of the network $f(\mathbf{x}_i) \in \mathbb{R}^{561}$ into logistic regression classifiers. Therefore, given tag labels $y_{it} \in \{-1, 1\}$ for song i and tag t , the network is trained to minimize the following loss:

$$\mathcal{L}_{\text{tag}} = \sum_{i,t} \log(1 + \exp(-y_{it} f_t(\mathbf{x}_i)))$$

Here we use a network with three fully-connected hidden layers and ReLU activations with dropout. Each layer has 1,200 neurons. Stochastic gradient descent with mini-batch of size 100 is used with AdaGrad [3] for adjusting the learning rate³. We notice that both dropout and AdaGrad are crucial for getting the good performance. The tagging performance is reported in Section 4.1.

3.2 Content-aware collaborative filtering

We can interpret the output of the last hidden layer $\mathbf{h}_i \in \mathbb{R}^{F_h}$ (here $F_h = 1200$) as a latent content representation of song i . Because of the way the network is trained, this latent representation is supposed to be highly correlated to the semantic tags (“topics” of music). Therefore, we can take a similar approach to the collaborative topic model and use this representation in a collaborative filtering model.

The generative process for the proposed model is as follows:

- For each user u , draw user latent factor:

$$\theta_u \sim \mathcal{N}(0, \lambda_\theta^{-1} I_K).$$

- For each song i , draw song latent factor:

$$\beta_i \sim \mathcal{N}(W\mathbf{h}_i, \lambda_\beta^{-1} I_K).$$

- For each user-song pair (u, i) , draw implicit feedback (whether user u listened to song i):

$$r_{ui} \sim \mathcal{N}(\theta_u^T \beta_i, c_{ui}^{-1}).$$

³The source code for training the neural network is available at: https://github.com/dawenl/deep_tagging

Here the weight matrix $W \in \mathbb{R}^{K \times F_h}$ transforms the learned content representation from the neural networks into the collaborative filtering latent space via Wh_i . The precision parameter λ_β balances how the song latent vector β_i deviates from the content feature. We set the confidence c_{ui} in the same way as in Section 2.1.

We want to emphasize that our proposed model is *content-aware* instead of *content-based*. Just like collaborative topic model, our proposed model is still fundamentally based on collaborative filtering. The content model is only used as a prior and can be deviated if the model thinks it is necessary to explain the data.

For notational convenience, we define the concatenated user latent factors matrix $\Theta \triangleq [\theta_1 | \dots | \theta_U] \in \mathbb{R}^{K \times U}$ and song latent factors matrix $B \triangleq [\beta_1 | \dots | \beta_I] \in \mathbb{R}^{K \times I}$. We estimate the model parameters $\{\Theta, B, W\}$ via maximum *a posteriori*.

The complete log-likelihood is written as:

$$\begin{aligned} \mathcal{L} = & - \sum_{u,i} \frac{c_{ui}}{2} (r_{ui} - \theta_u^T \beta_i)^2 - \frac{\lambda_\theta}{2} \sum_u \theta_u^T \theta_u \\ & - \frac{\lambda_\beta}{2} \sum_i (\beta_i - Wh_i)^T (\beta_i - Wh_i) \end{aligned}$$

Take the gradient of the complete log-likelihood with respect to the model parameters and set it to 0, we can obtain the following closed-form coordinate updates:

$$\theta_u \leftarrow (BC_u B^T + \lambda_\theta I_K)^{-1} BC_u r_u \quad (1)$$

$$\beta_i \leftarrow (\Theta C_i \Theta^T + \lambda_\beta I_K)^{-1} (\Theta C_i r_i + \lambda_\beta Wh_i) \quad (2)$$

$$W^T \leftarrow (H^T H + \lambda_W I_{F_h})^{-1} H^T B^T \quad (3)$$

where $C_u \in \mathbb{R}^{I \times I}$ is a diagonal matrix with c_{ui} , $i = 1, \dots, I$ as its diagonal elements, and $r_u \in \mathbb{R}^I$ is the feedback for user u . C_i and r_i are similarly defined. $H \in \mathbb{R}^{I \times F_h}$ is the concatenated output from the last hidden layer $[h_1 | \dots | h_I]^T$. When updating W , we add a small ridge term λ_W to the diagonal of the matrix to regularize and avoid numerical problems when inverting. Alternating between updating Θ , B , and W , we are guaranteed to reach a stationary point of the complete log-likelihood.

The same technique used in [6] to speed up computation can be applied here. This enables us to apply our model to large-scale music corpus and user-item interaction, which is not possible for CTR.

After the model is trained, we can make *in-matrix* prediction by $\hat{r}_{ui} = \theta_u^T \beta_i$. Similar to the collaborative topic model, we can also make *out-of-matrix* prediction for songs that no one has listened to by only using the content $\hat{r}_{ui} = \theta_u^T (Wh_i)$.

4. EVALUATION

We first evaluate our system on the pre-training tag prediction task to ensure the quality of the extracted features, and then measure its recommendation performance in comparison with related models⁴.

⁴ https://github.com/dawenl/content_wmf contains the source code for training the proposed model and reproducing the experi-

Model	Prec	Recall	F-score	AROC	MAP
SPMF	0.127	0.146	0.136	0.712	0.120
NNet	0.184	0.207	0.195	0.781	0.178

Table 1: Annotation and retrieval performance on the Million Song Dataset from Poisson matrix factorization with stochastic inference (SPMF) [10] and the pre-trained neural network (NNet) described in Section 3.1. The standard error is on the order of 0.01, thus not included here.

4.1 Tag prediction

Evaluation tasks and metrics We evaluate the pre-trained neural network on semantic tags with an annotation task and a retrieval task. We use the same dataset in Liang *et al.* [10] from the Million Song Dataset [1] and compare with their result which, to our knowledge, is the state-of-the-art performance on large-scale tag prediction. Note that we only use tag prediction as a proxy to measure the quality of the content model and do not argue for our approach as an optimal one to automatic music tagging.

For the annotation task we seek to automatically tag unlabeled songs. To evaluate the model's ability to annotate songs, we compute the average per-tag precision, recall, and F-score on the held-out test set. For the retrieval task, given a query tag we seek to provide a list of songs which are related to that tag. To evaluate retrieval performance, for each tag in the vocabulary we ranked each song in the test set by the predicted probability. We then calculate the area under the receiver-operator curve (AROC) and mean average precision (MAP) for each ranking.

Tagging performance and discussion The results are reported in Table 1, which show that the pre-trained neural network performs significantly better than the Poisson-factorization-based approach. This is not surprising for two reasons: 1) Here we treat tag prediction as a supervised task and train a multi-layer neural network, while in [10] the problem is formulated as an unsupervised learning task to account for the uncertainty in the user-generated tags (which incidentally can be considered as a typical example of implicit feedback). 2) Similar to LDA, Poisson factorization can only capture linear factor, whose expressive power is much weaker than that of a multi-layer neural network.

Nevertheless, the results confirm that our pre-trained neural network can be considered as an effective content feature extractor and we will use the output of the last hidden layer as the content feature.

Note that our neural network is relatively simple and does not directly use raw acoustic features (e.g., log-mel spectrograms) as input. It is reasonable to believe that with a more complex network structure and low-level acoustic feature, we should be able to achieve better tagging performance and obtain a more powerful content feature extractor, which could further boost the performance of our proposed recommendation method.

mental results for recommendation in Section 4.2.

Model	$R@40$	$R@80$	$R@120$	$R@160$	$R@200$	NDCG
PMF [4]	0.1021	0.1533	0.1908	0.2206	0.2456	0.2419
CTPF [5]	0.1031	0.1511	0.1861	0.2138	0.2370	0.2395
WMF [6]	0.1722	0.2367	0.2803	0.3133	0.3397	0.2881
CF + shallow	0.1724	0.2368	0.2803	0.3131	0.3396	0.2883
CF + deep	0.1722	0.2365	0.2800	0.3129	0.3394	0.2882

Table 2: *In-matrix* performance on the DEN subset with proposed and competing methods.

4.2 Recommendation

Data preparation We use the Taste Profile Dataset which is part of the Million Song Dataset to evaluate the recommendation performance. It contains listening history in the form of play counts from one million users with more than 40 million (user, song, play count) triplets. We first binarize all the play counts⁵ and create two complementary subsets (denoted as *DEN* and *SPR*):

For the DEN subset, we intend to create a reasonably dense subset so that the traditional collaborative filtering model will have good performance. We remove the users who have less than 20 songs in their listening history and songs that are listened to by less than 50 users, obtaining a subset with 613,682 users and 97,414 songs with more than 38 million user-song pairs (sparsity level 0.064%). For the SPR subset, on the contrary, we only keep the users who have less than 20 songs in their listening history and songs that are listened to by less than 50 users, yielding a highly sparse (0.002%) subset with 564,437 users and 260,345 songs.

We select 5% of the songs from DEN (4,871) for *out-of-matrix* prediction. For both subsets we split 20% and 10% as test and validation sets, respectively. Validation set is used to select hyperparameters, as well as monitor convergence by computing predictive likelihood.

Competing methods We compare our proposed method (denoted as *CF + deep*) with weighted matrix factorization (*WMF*) [6], as well as the following three methods:

CF + shallow: A simple baseline where we directly use the normalized VQ feature \mathbf{x}_i in place of the feature extracted from the neural network \mathbf{h}_i . This baseline is mainly used to demonstrate the necessity of an effective feature extractor for *out-of-matrix* prediction.

Poisson matrix factorization (*PMF*) [4]: Just like WMF, PMF is a matrix factorization model for collaborative filtering. Instead of Gaussian likelihood and priors on the latent factors, it utilizes Poisson likelihood model and gamma priors. The biggest advantage of PMF is computational. As shown in [4], the inference algorithm has complexity that scales linearly with the number of non-zero entries in the user-item matrix.

Collaborative topic Poisson factorization (*CTPF*) [5]: This model incorporates the content information into PMF in the same way as CTR. Additionally, it is conditionally conjugate with closed-form updates and enjoys the same

⁵ In practice, we find that the performances using actual play counts and binarized indicators are very close for our model.

computational efficiency as PMF. Therefore, it can be applied to large-scale dataset without delicate engineering.

Based on our argument in Section 2.3, we do not directly compare with [13] because it is sufficient to compare with WMF. For *out-of-matrix* recommendation evaluation, we can only compare with CTPF and CF + shallow. In all the experiments, the dimensionality of the latent space $K = 50$. We select $\alpha = 2$ and $\epsilon = 10^{-6}$ to compute the confidence c_{ui} . For WMF, CF + shallow, and CF + deep, the model parameters Θ , B and W (if any) are initialized to the same values.

Evaluation metrics To evaluate different algorithms, we produce a ranked list of all the songs (excluding those in the training and validation sets) for each user based on the predicted preference \hat{r}_u .

Precision and recall are commonly used evaluation metrics. However, for implicit feedback, the zeros can mean either the user is not interested in the song or more likely, the user does not know the song. This makes the precision less interpretable. However, since the non-zero r_{ui} 's are known to be true positive, we instead report *Recall@M*, which only considers songs within the top M in the ranked list. For each user, the definition of *Recall@M* is

$$Recall@M = \frac{\# \text{ songs that the user listened to in top } M}{\text{total } \# \text{ songs the user has listened to}}.$$

In addition to *Recall@M*, we also report (untruncated) normalized discounted cumulative gain (NDCG) [7]. Unlike *Recall@M* which only focuses on top M songs in the predicted list, NDCG measures the global quality of recommendation. In the meantime, it also prefers algorithms that place held-out test items higher in the list by applying a discounted weight. Given a ranked list of songs from the recommendation algorithm, for each user NDCG can be computed as follows:

$$DCG = \sum_{i=1}^I \frac{2^{rel_i} - 1}{\log_2(i + 1)}; \quad NDCG = \frac{DCG}{IDCG}.$$

Given our binarized data, the relevance rel_i is also binary: 1 if song i is in the held-out user listening history and 0 otherwise. IDCG is the optimal DCG score where all the held-out test songs are ranked top in the list. Therefore, larger NDCG values indicate better performance.

Results on the DEN subset The model hyperparameters $\lambda_\theta = \lambda_W = 10$ and $\lambda_\beta = 100$ are selected from the validation set based on NDCG. The *in-matrix* and *out-of-matrix* performances are reported in Table 2 and 3, respectively.

Model	$R@40$	$R@80$	$R@120$	$R@160$	$R@200$	NDCG
CTPF [5]	0.0256	0.0700	0.1440	0.1869	0.2086	0.1271
CF + shallow	0.0503	0.0894	0.1218	0.1514	0.1778	0.1429
CF + deep	0.0910	0.1461	0.1881	0.2241	0.2550	0.1605

Table 3: *Out-of-matrix* performance on the DEN subset with proposed and competing methods.

Model	$R@40$	$R@80$	$R@120$	$R@160$	$R@200$	NDCG
WMF [6]	0.1137	0.1286	0.1378	0.1449	0.1505	0.1415
CF + shallow	0.1138	0.1286	0.1377	0.1449	0.1504	0.1416
CF + deep	0.1140	0.1289	0.1378	0.1451	0.1507	0.1417

Table 4: *In-matrix* performance on the SPR subset with proposed and competing methods.

All the metrics are averaged across 612,232 users in the held-out test user-item pairs.

We can see that with sufficient amount of user feedback, there is almost no difference in performance among WMF, CF + shallow, and CF + deep⁶ – there is not a single model which is consistently better. This is understandable, since both CF + shallow and CF + deep are fundamentally collaborative filtering models. With enough user feedback, the model is able to produce meaningful recommendation without resorting to the content features. Moreover, CF + shallow, which has access to more content information, does slightly better than CF + deep.

One observation from Table 2 is that adding content features does not necessarily improvement the performance. Unlike CF + deep, CTPF falls behind its content-free counterpart PMF on both $Recall@M$ and NDCG. This is possibly due to the insufficient feature extraction capability of the topic model (LDA) on the rich musical data.

The superiority of CF + deep is more obvious on the *out-of-matrix* predictions performance shown in Table 3. We can see a larger margin between CF + deep and CF + shallow, as compared to their close performance on *in-matrix* predictions. This suggests the importance of a powerful feature extractor in the absence of usage data. Even a simple linear LDA model in CTPF can be more effective than CF + shallow at predicting songs that the users listened to in the held-out test set.

Results on the SPR subset We repeat the *in-matrix* evaluation on the highly sparse SPR subset. The model hyperparameters $\lambda_\theta = \lambda_W = 10^{-2}$ and $\lambda_\beta = 1$ are selected from the validation set. The performance is reported in Table 4. All the metrics are averaged across 564,437 users in the held-out test user-item pairs.

Again, the overall differences among all three methods are relatively minor. However, with very limited user feedback, both CF + shallow and CF + deep outperform the content-free WMF. More importantly, CF + deep consistently improves over CF + shallow, which indicates the importance of an effective feature extractor.

⁶ There is little point in arguing for the statistical significance of the difference, since given the number of users to average over, the standard error is vanishingly small.

5. CONCLUSION

In this paper we present a content-aware collaborative music recommendation system that joins a multi-layer neural network content model with a collaborative filtering model. The system achieves the state-of-the-art performance in music recommendation given content and implicit feedback data.

A possible future direction is to incorporate ranking-based loss function, e.g., the weighted approximate-rank pairwise (WARP) loss in [16] into the collaborative filtering model. We normally evaluate recommendation algorithms using ranking-based metrics (e.g. $Recall@M$ and NDCG), but the model is trained using squared loss function. It would be more natural to directly optimize a ranking-based loss function.

6. ACKNOWLEDGEMENTS

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Tesla K40 GPU used for this research.

7. REFERENCES

- [1] Thierry Bertin-Mahieux, Daniel P.W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In *ISMIR*, 2011.
- [2] David M. Blei, Andrew Y. Ng, and Michael I. Jordan. Latent Dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.
- [3] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.
- [4] Prem Gopalan, Jake M. Hofman, and David M. Blei. Scalable recommendation with poisson factorization. *arXiv preprint arXiv:1311.1704*, 2013.
- [5] Prem K. Gopalan, Laurent Charlin, and David Blei. Content-based recommendations with Poisson factorization. In *Advances in Neural Information Processing Systems 27*, pages 3176–3184. 2014.

- [6] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. IEEE, 2008.
- [7] Kalervo Järvelin and Jaana Kekäläinen. Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*, 20(4):422–446, 2002.
- [8] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.
- [9] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.
- [10] Dawen Liang, John Paisley, and Daniel P. W. Ellis. Codebook-based scalable music tagging with Poisson matrix factorization. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 167–172, 2014.
- [11] Brian McFee, Luke Barrington, and Gert Lanckriet. Learning similarity from collaborative filters. In *ISMIR*, pages 345–350, 2010.
- [12] Andriy Mnih and Ruslan Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.
- [13] Aäron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In *Advances in Neural Information Processing Systems*, pages 2643–2651, 2013.
- [14] Chong Wang and David M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 448–456. ACM, 2011.
- [15] Xinxi Wang and Ye Wang. Improving content-based and hybrid music recommendation using deep learning. In *Proceedings of the 22nd ACM International Conference on Multimedia*. ACM Press, 2014.
- [16] Jason Weston, Samy Bengio, and Nicolas Usunier. Ws-abc: Scaling up to large vocabulary image annotation. In *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI*, 2011.
- [17] Kazuyoshi Yoshii, Masataka Goto, Kazunori Komatani, Tetsuya Ogata, and Hiroshi G. Okuno. Hybrid collaborative and content-based music recommendation using probabilistic model with latent user preferences. In *ISMIR*, 2006.

COMPARATIVE ANALYSIS OF ORCHESTRAL PERFORMANCE RECORDINGS: AN IMAGE-BASED APPROACH

Cynthia C. S. Liem

Delft University of Technology
Multimedia Computing Group
c.c.s.liem@tudelft.nl

Alan Hanjalic

Delft University of Technology
Multimedia Computing Group
a.hanjalic@tudelft.nl

ABSTRACT

Traditionally, the computer-assisted comparison of multiple performances of the same piece focused on performances on single instruments. Due to data availability, there also has been a strong bias towards analyzing piano performances, in which local timing, dynamics and articulation are important expressive performance features. In this paper, we consider the problem of analyzing multiple performances of the same symphonic piece, performed by different orchestras and different conductors. While differences between interpretations in this genre may include commonly studied features on timing, dynamics and articulation, the timbre of the orchestra and choices of balance within the ensemble are other important aspects distinguishing different orchestral interpretations from one another. While it is hard to model these higher-level aspects as explicit audio features, they can usually be noted visually in spectrogram plots. We therefore propose a method to compare orchestra performances by examining visual spectrogram characteristics. Inspired by eigenfaces in human face recognition, we apply Principal Components Analysis on synchronized performance fragments to localize areas of cross-performance variation in time and frequency. We discuss how this information can be used to examine performer differences, and how beyond pairwise comparison, relative differences can be studied between multiple performances in a corpus at once.

1. INTRODUCTION

A written notation is not the final, ultimate representation of music. As Babbitt proposed, music can be represented in the acoustic (physical), auditory (perceived) and graphemic (notated) domain, and as Wiggins noted, in each of these, projections are observed of the abstract and intangible concept of ‘music’ [29]. In classical music, composers usually write down a notated score. Subsequently, in performance, multiple different musicians will present their own artistic reading and interpretation of it.

Nowadays, increasing amounts of digital music recordings become available. As a consequence, for musical pieces, an increasing amount of (different) recorded performances can be found. Therefore, in terms of data availability, increasing opportunities emerge to study and compare different recordings of the same piece. Beyond the Music Information Retrieval (Music-IR) domain, this can serve long-term interests in psychology and cognition on processes and manifestations of expressive playing (e.g. [6, 21, 26]), while the analysis of performance styles and schools also is of interest to musicologists [5, 16].

In this paper, we mostly are interested in the analysis of multiple performances of the same piece from a search engine and archive exploration perspective. If one is looking for a piece and is confronted with multiple alternative performances, how can technology assist in giving overviews of main differences between available performances? Given a corpus, are certain performances very similar or dissimilar to one another?

In contrast to common approaches in automated analysis of multiple performances, we will not depart from explicit modeling of performance parameters from a signal. Instead, we take a more holistic approach, proposing to consider spectrogram images. This choice has two reasons: first of all, we are particularly interested in finding methods for comparative analysis of orchestra recordings. We conjecture that the richness of orchestra sounds is better captured in spectrogram images than in mid-level audio features. Secondly, as we will demonstrate in this paper, we believe spectrogram images offer interpretable insights into performance nuances.

After discussing the state of the art in performance analysis in Section 2, in Section 3, we will further motivate our choice to compare performances through visual comparison of spectrogram images. Subsequently, Section 4 details our chosen comparison method, after which we present the experimental setup for this paper in Section 5. We will then illustrate our approach and its outcomes through a case study in Section 6, with a detailed discussion of selected musically meaningful examples. This is followed by a discussion on how our method can assist corpus-wide clustering of performances in Section 7, after which the Conclusion will be presented.



© Cynthia C. S. Liem, Alan Hanjalic.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Cynthia C. S. Liem, Alan Hanjalic. “Comparative analysis of orchestral performance recordings: an image-based approach”, 16th International Society for Music Information Retrieval Conference, 2015.

2. STATE-OF-THE-ART REVIEW

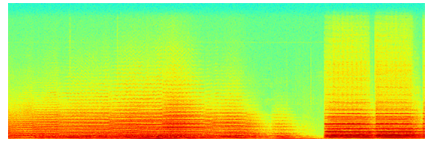
A lot of work exists on analyzing musical performance expressivity. In several cases, establishing models for computer-rendered expressive performances was the ultimate goal (e.g. see [10, 11]). Other works focused on identifying reasons behind performance expressivity, including lower-level perceptual processes [21]; varying score editions, individual treatments of ornamentation and pedaling, and music-theoretic notions of expectation and tension-relaxation [20]; generative rules, emotional expression, random variability, motion principles and stylistic unexpectedness [14]; and musical structure [9, 13, 20]. Historically, the analysis of musical performance strongly focused on expressivity in piano playing (e.g. [6, 20–22]). The few exceptions to this rule focused on violin performance (e.g. [4]), movement in clarinet players (e.g. [8]), and performance of trained and untrained singers (e.g. [7], inspired by [26]), but to the best of our knowledge, no systematic comparative studies have been performed considering larger ensembles.

A reason for the general bias towards piano performance may be that digital player pianos (e.g. the Yamaha Disklavier) allow a very precise recording of mechanical performance parameters. When such parameters are available, inter-onset-intervals (IOIs), expressing the time between subsequent onsets, are frequently studied. Otherwise, performance parameters have to be extracted or annotated from the audio signal. As a piano has a discrete pitch set and percussive mechanics, expressive possibilities for a pianist are restricted to timing, dynamics and articulation. As a consequence, audio-based performance analysis methods usually focus on local timing and dynamics. Since it is not trivial to find a suitable time unit for which these parameters should be extracted, supervised or semi-supervised methods often have been applied to obtain this, e.g. by departing from manually annotating beat labels (e.g. [24, 25]). However, it is hard (if not infeasible) to realize such a (semi-)supervised approach at scale. Therefore, while a very large corpus of recorded Chopin Mazurkas exists, in practice only the Mazurkas for which annotated beat information exists have been studied in further depth (e.g. [15, 19, 24, 25]).

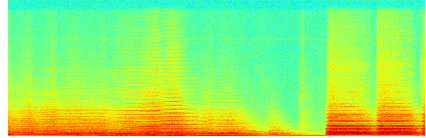
Alternatively, in [17, 18] an unsupervised approach for comparing Mazurka recordings was proposed which does not rely on explicitly modeled higher-level performance parameters or semantic temporal units, but rather on alignment patterns from low-level short-time frame analyses. As such, this approach would be scalable to a larger corpus. Furthermore, while the choice of not adopting explicit performance parameters makes evaluation of a clear-cut ground truth less trivial, at the same time it allows for any salient variations to emerge automatically from the analysis. The work of this paper follows a similar philosophy.

3. MOTIVATION FOR SPECTROGRAM IMAGES

In this paper, we focus on the comparative analysis of orchestra recordings. An orchestra involves a mix of many



(a) Georg Solti, Chicago Symphony Orchestra, 1973.



(b) Nikolaus Harnoncourt, Chamber Orchestra of Europe, 1990.

Figure 1. Beethoven's *Eroica* symphony, 2nd movement, spectrogram of bars 56-60 for two different interpretations.

instruments. Hence, the overall orchestral sound is richer than that of a piano, although individual beat placings and note onsets will be much smoother. Given the multitude of involved players, an orchestra needs guidance by a conductor. Due to this coordinated setup, there is less room for individual freedom in both local dynamics and tempo than in Romantic piano music repertoire. Thus, while local tempo deviations still occur in orchestral recordings, one cannot expect these to reflect performer individuality as strongly as for example in the case of Chopin Mazurkas.

At the same time, in terms of timbre, balance and phrasing articulation, a conductor has a much richer palette than isolated instruments can offer. These aspects are not trivial to explicitly model or interpret from audio signals. However, relevant information may be reflected in recording spectrograms, as illustrated in Figure 1. While it is hard to point out individual instruments, a spectrogram can visually reveal how rich the overall sound is, where signal energy is concentrated, and if there are any salient sound quality developments over time, such as vibrato notes.

Indeed, spectrograms are commonly used in audio editing tools for visualization, navigation and analysis purposes. In an ethnographic study of musicologists studying historical recordings, it further was shown that examination of the spectrogram helped musicologists in discovering and listening to performance nuances [1]. Therefore, regarding potential end users of performance analysis and exploration tools, spectrogram images may be more familiar and interpretable than reduced mid-level representations such as chroma.

4. METHOD

Our proposed analysis method for spectrogram images is inspired by the eigenfaces method of Turk and Pentland [27], which was originally proposed in the context of human face recognition. Since human faces share many common features, by applying Principal Components Analysis (PCA) on a dataset of aligned facial im-

ages, a set of basis images ('eigenfaces') can be found, explaining most of the variability found in the face dataset. While PCA has previously been applied as a tool in musical performance analysis [23], this analysis was performed on annotation-intensive IOI data. In contrast, our analysis considers information which only requires alignment of different fragments (as will be described in Section 5), but no further manual annotation effort.

We apply the same principle to a set of N spectrogram images for a time-aligned music fragment, as represented by N different recordings. Each spectrogram image \mathbf{x} is $(i \cdot j)$ pixels in size. We treat each pixel in the image as a feature; as such, \mathbf{x} is a vector of length $i \cdot j$. We collect all spectrogram images in an $(N \times (i \cdot j))$ matrix \mathbf{X} .

By applying PCA, we decompose \mathbf{X} into an $(N \times N)$ matrix of principal component loadings \mathbf{W} and an $((i \cdot j) \times N)$ matrix of principal components scores \mathbf{T} . \mathbf{X} can be reconstructed by performing $\mathbf{X} = \mathbf{T} \cdot \mathbf{W}^T$.

Since the PCA is constructed such that principal components are ordered in descending order of variance, dimension reduction can be applied by not using the full \mathbf{T} and \mathbf{W} , but only the first L columns of both.

The component scores in \mathbf{T} can now be interpreted and visualized as basis images, each representing a linear component explaining part of the variability in the dataset.

5. EXPERIMENTAL SETUP

Unfortunately, no standardized corpora on multiple performances of the same orchestra piece exist.¹ Furthermore, no clear-cut ground truth exists of performance similarity. We therefore consider a dataset collected for the PHENICX² project, consisting of 24 full-length recordings of Beethoven's Eroica symphony, as well as 7 recordings of the Alpensinfonie by Richard Strauss. In the Beethoven dataset, 18 different conductors and 10 orchestras are featured (with a major role for the recording catalogue of the Royal Concertgebouw Orchestra (RCO)), meaning that the same conductor may conduct multiple orchestras, or even the same orchestra at different recording moments. While metadata and audio content are not fully identical, in two cases in the dataset (Harnoncourt, Chamber Orchestra of Europe (COE) 1990 and 1991; Haitink, London Symphony Orchestra (LSO) 2005 ($\times 2$)), there are suspicions that these near-duplicates pairs consider the same original recording. In the Strauss dataset, 6 conductors and 6 orchestras are featured: Haitink conducts both the RCO and LSO, and the RCO is represented once more with Mariss Jansons as conductor. The oldest (Mengelberg, RCO, 1940) and newest (Fischer, RCO, 2013) recordings are both featured in the Beethoven dataset.

We will demonstrate insights from the PCA spectrogram analysis in two ways: (1) by highlighting several analysis examples in detail in Section 6, based on manual selection of musically relevant fragments and (2) by discussing generalization opportunities in Section 7, based on

¹ While a dataset of orchestral recordings with multiple renditions of the same piece was used in [2], these recordings are not publicly available.

² <http://phenicx.upf.edu>



Figure 2. Eroica 1st movement, score bars 3-10.

aggregation of 4-bar analysis frames.

In both cases, a similar strategy is taken: first, a musical fragment is designated, for which all recordings of the piece should be aligned. Alignment is performed automatically using the method described in [12]. Then, the audio fragments, which are all sampled at $F_s = 44.1$ kHz, are analyzed using a Hann window of 1024 samples and a hop size of 512, and the corresponding magnitude spectrum is computed using the Essentia framework [3]. Combining the spectra for all frames results in a spectrogram image. To ensure that all images have equal dimensions, a constant height of 500 pixels is imposed, and the longest fragment in terms of time determines a fixed width of the image, to which all other spectrograms are scaled accordingly. While all recordings are offered at 44.1 kHz, the original recordings sometimes were performed at a lower sampling rate (particularly in more historical recordings). Therefore, a sharp energy cut-off may exist in the higher frequency zones, and for analysis, we try to avoid this as much as possible by only considering the lower 90% of the image. In general, by using raw spectrogram images, a risk is that recording quality is reflected in this spectrum; nonetheless, in the next sections we will discuss how musically relevant information can still be inferred.

6. CASE STUDY

In this case study, to illustrate the information revealed by PCA analysis, we will look in detail at information obtained on two selected fragments: the start of the first movement of the Eroica symphony, first theme (bars 3-15), and the 'maggiore' part of the Eroica symphony, second movement (bars 69-104).

6.1 Eroica first movement, bars 3-15

A score fragment for bars 3-10 of the first movement of the Eroica is given in Figure 2. In our case, we consider the full phrase up to bar 15 in our analysis.

The first three basis images (component scores) resulting from PCA analysis are shown in Figure 3. The first component of the PCA analysis gives a smoothed 'basic' performance version of the fragment. For this very general component, it is rather hard to truly contrast performances. However, a more interesting mapping can be done in higher-order components. As an example, Figure 4 dis-

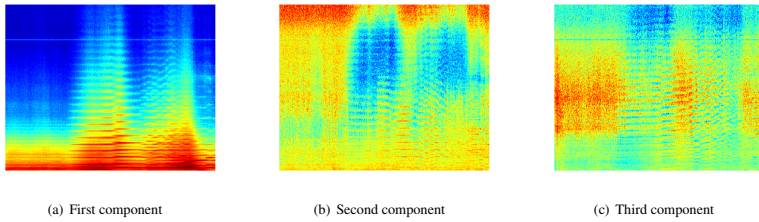


Figure 3. Eroica, 1st movement, 1st theme start (bars 3-15); first three principal component basis images.

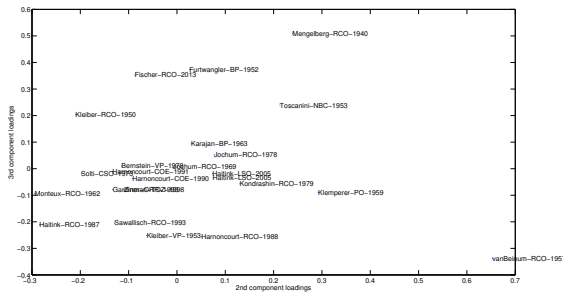


Figure 4. 2nd and 3rd PCA component scatter plot for Eroica 1st movement, bars 3-15.

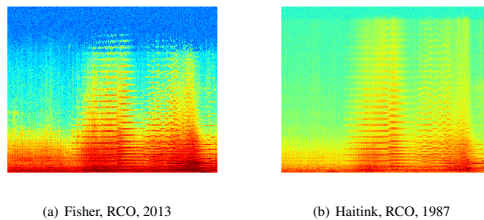


Figure 5. Spectrogram image examples for Fisher and Haitink interpretations of Eroica 1st movement, bars 3-15.

plays a scatter plot of the second and third principal component loadings for this fragment.

While as expected, several historical (and acoustically noisy) recordings cause outliers, by comparing the component scores and loadings to corresponding data samples, we still note interpretable differences. For example, the RCO recordings of Fischer and Haitink, of which respective spectrogram images for the excerpt are shown in Figure 5, have contrasting loadings on the third PCA component. Judging from the principal component image in Figure 3, this component indicates variability at the start of the fragment (when the cello play), and in between the fragments highlighted by the second component; more specif-

ically, a variability hotspot occurs at the sforzato in bar 10. When contrasting two opposite examplars in terms of scores, such as Fischer and Haitink, it can be heard that in the opening, Haitink emphasizes the lower strings more strongly than Fischer, while at the sforzato, Haitink strongly emphasizes the high strings, and lets the sound develop over the a-flat played by violin 1 in bar 10. Fischer maintains a ‘tighter’ sound over this sforzato.

6.2 Eroica second movement, maggiore

To illustrate findings on another manually selected (and slightly longer) relevant fragment, we also consider the

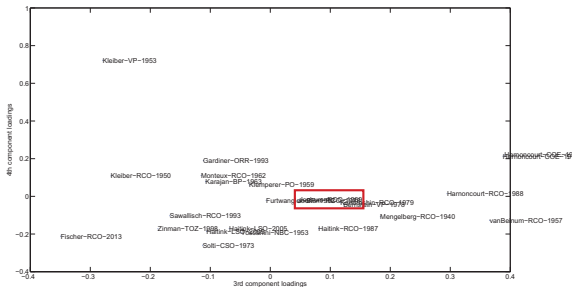


Figure 6. 3rd and 4th PCA component scatter plot for Eroica 2nd movement, maggiore. Jochum's 1969 and 1978 recordings occur within the marked rectangular border.

'maggiore' part of the second movement of the Eroica. Analyses of scatter plots and component images show that the second principal component is affected by historical recording artefacts. However, this is less so for the third and fourth component, of which the scatter plot is displayed in Figure 6. It can be seen that the suspected near-duplicates of Harmoncourt's two COE recordings have near-identical loadings on these components. Next to this, another strong similarity is noted between the recordings of Jochum with the RCO in 1969 and 1978. While these both recordings acoustically are clearly different and also seem to be explicitly different interpretations, there still are consistencies in Jochum's work with the same orchestra for these two recordings.

7. CORPUS-WIDE CLUSTERING

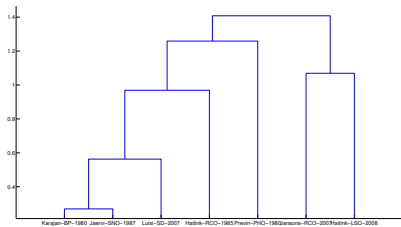
As demonstrated in the previous section, PCA analysis can be used as an exploratory tool to reveal differences between selected fragments in recordings. However, selecting incidental manual examples will not yet allow for scalable analysis of information over the full timeline of a piece. To do this, instead of pre-selecting designated fragments, we perform a 4-bar sliding window PCA analysis on full synchronized recordings, where bar boundaries are obtained through the score-to-performance mapping obtained in the alignment procedure. Instead of examining individual component images, in each 4-bar analysis frame, we consider vectors of component loadings for the minimum amount of components required to explain 95% of the variance observed. From these component loading vectors, we compute the Euclidean distance between recordings within a frame, and aggregate these at the recording track level.³

³Note that component loadings obtained for different frames cannot be directly averaged, as the components are different per frame. However, observed distances between recordings still remain valid and can be aggregated.

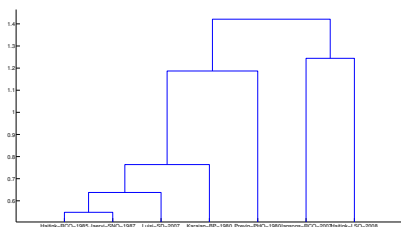
Based on distances found between performances, clustering can be performed. This reveals whether stable performer clusters can be found for different movements within a piece, and to what extent clusterings found in local fragments match those found for a full piece.

Regarding the first question, for each of the Eroica movements, we calculated the average between-performer distances per movement, and then made 5 clusters of performers based on Ward's linkage method [28]. While space does not allow a full cluster result report, several clusters co-occur consistently:

- The two Harmoncourt COE recordings consistently form a separate cluster. These are highly likely to be duplicate recordings.
- Haitink's two LSO recordings also consistently co-occur, and like Harmoncourt are highly likely to be duplicate recordings. However, Bernstein's 1978 Vienna Philharmonic recording co-occurs with these two Haitink recordings in the first three Eroica movements, and thus may be similar in terms of interpretation. It is striking that Haitink's 1987 recording with the RCO never co-occurs in this cluster.
- In the first three movements, a consistent cluster occurs with recordings by Kleiber (Philharmonia Orchestra, 1959), Toscanini (NBC Symphony Orchestra, 1953) and Van Beinum (RCO, 1957). While this may be due to recording artefacts, other historical recordings (e.g. Kleiber, RCO 1950 / Vienna Philharmonic 1953) do not co-occur.
- Surprisingly, Gardiner's historically informed recording with the Orchestre Révolutionnaire et Romantique (1993) clusters with Kleiber's 1950 RCO recording for the first and last movement of the Eroica. Upon closer listening, Gardiner's choice of concert pitch matches the pitch of Kleiber's recording, and the sound qualities of the orchestras



(a) 'Sonnenaufgang' fragment (bars 46-63).



(b) Average over full Alpensinfonie.

Figure 7. Dendrogram images for performer distances in the Alpensinfonie.

are indeed similar (although in case of Kleiber, this is caused by recording artefacts).

- The 1969 and 1978 Jochum recordings with the RCO always co-occur, though in the largest cluster of recordings. As such, they are similar, but no clear outlier pair compared to the rest of the corpus.

Regarding consistent clusterings over the course of a piece, we further illustrate an interesting finding from the Alpensinfonie, in which we compare a clustering obtained on 18 bars from the 'Sonnenaufgang' movement to the clustering obtained for average distances over the full piece, as visualized in the form of dendrograms in Figure 7. As can be noted, the clusterings are very close, with the only difference that within the 'Sonnenaufgang' movement, Karajan's interpretation is unusually close to Järvi's interpretation, while Haitink's interpretation is unusually different.

8. CONCLUSION

In this paper, we proposed to analyze differences between orchestral performance recordings through PCA analysis of spectrogram images. As we showed, PCA analysis is capable of visualizing areas of spectral variation between recordings. It can be applied in a sliding window setup to assess differences between performers over the timeline

of a piece, and findings can be aggregated over interpretations of multiple movements. While spectrograms inevitably have sensitivity to recording artefacts, we showed that near-duplicate recordings in the corpus could be identified, and historical recordings in the corpus do not consistently form outliers in the different analyses.

While certain interesting co-occurrences were found between recordings, no conclusive evidence was found regarding consistent clustering of the same conductor with different orchestras, or the same orchestra with different conductors. This can either be due to interference from artefacts and different recording setups, but at the same time may suggest that different conductors work differently with different orchestras.

Several directions of future work can be identified. First of all, further refinement regarding the generation and analysis of the spectrogram images should be performed. At the moment, given the linear way of plotting and high sample rate, the plain spectrogram may be biased towards higher-frequency components, and risks to be influenced by sharp frequency cut-offs from lower original recording sample rates.

Furthermore, it would be interesting to study more deeply if visual inspection of spectrograms can indeed assist people in becoming more actively aware of performance differences. While the spectrogram images are expected to already be understandable to potential end-users, appropriate techniques should still be found for visualizing differences between multiple performers in a corpus. In the current paper, this was done with scatter plots and dendrograms, but for non-technical end-users, more intuitive and less mathematically-looking visualizations may be more appropriate.

One concern that may come up with respect to our work, is that it may be hard to fully associate our reported findings to expressive performance. As indicated, recording artefacts are superimposed on the signal, and effects of different halls and choices of orchestra instruments and concert pitch may further influence acoustic characteristics, which will in turn influence our analysis. Furthermore, since we are dealing with commercial recordings, we are dealing with produced end results which may have been formed out of multiple takes, and as such do not reflect 'spontaneous' performance.

However, our main interest is not in analyzing performance expressivity per se, but in providing novel ways for archive and search engine exploration, and making general sense of larger volumes of unannotated performance recordings. In such settings, the data under study will mostly be produced recordings with the above characteristics. For this, we believe our approach is useful and appropriate, offering interesting application opportunities.

Acknowledgements: The research leading to these results has received funding from the European Union Seventh Framework Programme FP7 / 2007–2013 through the PHENICX project under Grant Agreement no. 601166.

9. REFERENCES

- [1] M. Barthet and S. Dixon. Ethnographic observations of musicologists at the British Library: implications for Music Information Retrieval. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Miami, USA, 2011.
- [2] J. P. Bello. Measuring structural similarity in music. *IEEE Transactions on Audio, Speech and Language Processing*, 19(7):2013–2025, 2011.
- [3] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra. ESSENTIA: an Audio Analysis Library for Music Information Retrieval. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 493–498, 2013.
- [4] E. Cheng and E. Chew. Quantitative Analysis of Phrasing Strategies in Expressive Performance: Computational Methods and Analysis of Performances of Unaccompanied Bach for Solo Violin. *Journal of New Music Research*, 37:325–338, December 2008.
- [5] N. Cook. Towards the compleat musicologist? In *Proceedings of the International Symposium on Music Information Retrieval (ISMIR) [invited talk]*, London, UK, 2005.
- [6] P. Desain and H. Honing. Does expressive timing in music performance scale proportionally with tempo? *Psychological Research*, 56(4):285–292, July 1994.
- [7] J. Devaney, M. I. Mandel, D. P. W. Ellis, and I. Fujinaga. Automatically extracting performance data from recordings of trained singers. *Psychomusicology: Music, Mind & Brain*, 21:108–136, 2011.
- [8] M. M. Wanderley E. C. F. Teixeira, M. A. Loureiro and H. C. Yehia. Motion Analysis of Clarinet Performers. *Journal of New Music Research*, July 2014.
- [9] A. Friberg and J. Sundberg. Does music performance allude to locomotion? A model of final *ritardandi* derived from measurements of stopping runners. *Journal of the Acoustic Society of America*, 105(3):1469–1484, March 1999.
- [10] W. Goebel, S. Dixon, G. De Poli, A. Friberg, R. Bresin, and G. Widmer. “Sense” in expressive music performance: Data acquisition, computational studies, and models. In P. Polotti and D. Rocchesso, editors, *Sound to sense, sense to sound: a state of the art in sound and music computing*. Logos Verlag, 2007.
- [11] W. Goebel and G. Widmer. On the use of computational methods for expressive music performance. In T. T. Crawford and L. Gibson, editors, *Modern Methods for Musicology: Prospects, Proposals and Realities*. Digital Research in the Arts and Humanities, pages 93–113. Ashgate, 2009.
- [12] M. Grachten, M. Gasser, A. Arzt, and G. Widmer. Automatic Alignment of Music Performances with Structural Differences. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, pages 607–612, 2013.
- [13] M. Grachten and G. Widmer. Who is who in the end? Recognizing pianists by their final *ritardandi*. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Kobe, Japan, October 2009.
- [14] P. N. Juslin. Five facets of musical expression: a psychologist’s perspective on music performance. *Psychology of Music*, 31(3):273–302, July 2003.
- [15] K. Kosta, O. F. Bandtlow, and E. Chew. Practical implications of dynamic markings in the score: Is piano always piano? In *Proceedings of the 53rd International AES Conference on Semantic Audio*, London, UK, January 2014.
- [16] E. Lieberman, E. Ernoy, and B. Chor. A Phylogenetic Approach to Music Performance Analysis. *Journal of New Music Research*, 41:195–222, June 2012.
- [17] C. C. S. Liem and A. Hanjalic. Expressive timing from cross-performance and audio-based alignment patterns: An extended case study. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Miami, Florida, USA, October 2011.
- [18] C. C. S. Liem, A. Hanjalic, and C. S. Sapp. Expressivity in musical timing in relation to musical structure and interpretation: A cross-performance, audio-based approach. In *Proceedings of the 42nd International AES Conference on Semantic Audio*, pages 255–264, Ilmenau, Germany, July 2011.
- [19] M. Müller, P. Grosche, and C. S. Sapp. What makes beat tracking difficult? a case study on chopin mazurkas. In *Proceedings of the International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, The Netherlands, August 2010.
- [20] C. Palmer. Anatomy of a performance: Sources of musical expression. *Music Perception*, 13:433–453, Spring 1996.
- [21] A. Penel and X. Drake. Sources of timing variations in music performance: a psychological segmentation model. *Psychological Research*, 61(1):12–32, March 1998.
- [22] B. Repp. A microcosm of musical expression. I. Quantitative analysis of pianist’s timing in the initial measures of Chopin’s Etude in E major. *Journal of the Acoustic Society of America*, 104(2):1085–1100, August 1998.
- [23] B. Repp. A microcosm of musical expression. I. Quantitative analysis of pianist’s timing in the initial measures of Chopin’s Etude in E major. *Journal of the Acoustic Society of America*, 104(2):1085–1100, August 1998.
- [24] C. S. Sapp. Comparative analysis of multiple musical performances. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Vienna, Austria, September 2007.
- [25] C. S. Sapp. Hybrid numeric/rank similarity metrics for musical performance analysis. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Philadelphia, USA, September 2008.
- [26] C. Seashore. *Psychology of music*. University of Iowa Press, Iowa City, 1938.
- [27] M. A. Turk and A. P. Pentland. Face recognition using eigenfaces. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Maui, Hawaii, USA, June 1991.
- [28] J. H. Ward Jr. Hierarchical grouping to optimize an objective function. *Journal of the American Statistical Association*, 58(301):236–244, 1963.
- [29] G. A. Wiggins. Computer-representation of music in the research environment. In T. T. Crawford and L. Gibson, editors, *Modern Methods for Musicology: Prospects, Proposals and Realities*. Digital Research in the Arts and Humanities, pages 7–22. Ashgate, Aldershot, UK, 2009.

MONAURAL BLIND SOURCE SEPARATION IN THE CONTEXT OF VOCAL DETECTION

Bernhard Lehner, Gerhard Widmer
Department of Computational Perception
Johannes Kepler University of Linz
{bernhard.lehner, gerhard.widmer}@jku.at

ABSTRACT

In this paper, we evaluate the usefulness of several monaural blind source separation (BSS) algorithms in the context of vocal detection (VD). BSS is the problem of recovering several sources, given only a mixture. VD is the problem of automatically identifying the parts in a mixed audio signal, where at least one person is singing. We compare the results of three different strategies for utilising the estimated singing voice signals from four state-of-the-art source separation algorithms. In order to assess the performance of those strategies on an internal data set, we use two different feature sets, each fed to two different classifiers. After selecting the most promising approach, the results on two publicly available data sets are presented. In an additional experiment, we use the improved VD for a simple post-processing technique: For the final estimation of the source signals, we decide to use either silence, or the mixed, or the separated signals, according to the VD. The results of traditionally used BSS evaluation methods suggest that this is useful for both the estimated background signals, as well as for the estimated vocals.

1. INTRODUCTION

Monaural Blind Source Separation (BSS) is a technique for the separation of at least two components from a single-channel signal without using additional information, like the instrumentation or the notation of a musical piece. It is extremely challenging, since we have to deal with the fact, that less mixtures than sources are at hand.

The result of BSS could be useful for many tasks like remixing, creating karaoke songs, manipulate isolated instruments, and so on. Certain Music Information Retrieval (MIR) tasks could also benefit from a BSS as a pre-processing step, e.g. vocalist similarity, pitch detection, automatic transcription, keyword spotting, ...

Unfortunately, it is hard to estimate the usefulness of a certain BSS algorithm for a specific task beforehand. Metrics usually used for evaluating BSS (see Section 4.1)

are certainly useful for comparison purposes, but have only limited meaningfulness when it comes to the ultimate question if BSS is actually useful for a specific task.

To give an example, Schuller et al. had no success with achieving better results for tempo and key detection by utilising drum beat separation in [20], despite the fact that the audible results seemed good enough to be used for music remixing. On the other hand, Weninger et al. achieved a significant performance gain in the 3-class task of detecting singing voice segments and simultaneously recognising the vocalist gender in [21].

Therefore, we evaluate the usefulness of several state-of-the-art BSS algorithms in the context of vocal detection (VD), also referred to as singing voice detection. For this task, usually several features are extracted frame-wise from the audio signal and fed to a classifier [9, 13, 14, 19, 21, 22], or even to a speech-recogniser [1] in order to obtain the vocal/non-vocal decision. Given this use case, we are mainly interested in separating the signal into two sources: vocals and background. In order to find the best usage of the BSS results, we discuss the outcome of three different strategies.

Furthermore, we investigate if the quality of the separated sources can be improved by using VD as a post-processing technique: For the estimated vocals we mute the parts which are not classified as such by our VD to reduce non-voiced artifacts. For the estimated background, we replace the parts which are classified as non-vocal by our VD with the original mixed audio signal.

2. SELECTED BSS ALGORITHMS

We selected four state-of-the-art BSS algorithms, all of them were already used to extract the singing voice from a mixed audio signal, and reference implementations are provided by the authors. Due to limited space, we can discuss the methods only briefly, and refer to the original papers.

The adaptive REpeating Pattern Extraction Technique (aREPET) is a method, where repeating patterns (background) are identified and used to separate non-repeating (foreground) elements. Those elements are often the varying vocals, and it was shown in [18], that this technique can be used for Music/Voice Separation. There are several variants of the REPET algorithm [11, 16–18], whereas according to the results from Liutkus et al. in [12] the



aREPET yielded the best Δ_{SDR} for vocals out of three variants. Therefore, we consider the aREPET the most promising variant and choose this for our comparison.

The FASST toolbox by Ozerov et al. [15] allows to specify prior information and implement arbitrary separation problems. Therefore, it is not merely a method, but more a general framework. However, a baseline implementation is included in the toolbox, which separates a song into the four sources drums, bass, main melody, and the rest. It comes with pre-trained models for several sources, incl. singing voice, which is in our case used to extract the main melody source.

The Kernel Additive Modelling (KAM) approach [10, 12] uses source-dependent proximity kernels to describe local dynamics like periodicity (similar to REPET), smoothness, stability over time or frequency, and more. The different sources are then separated by an algorithm called iterative kernel backfitting.

Huang et al. use in [8] Robust Principal Component Analysis (RPCA) for the separation of singing voice. Their basic assumptions are, that singing voice is relatively sparse within songs, and accompaniment is in a low-rank subspace due to its repetitive structure. Their method uses solely the spectrogram as input, and neither training nor particular features are required.

Another interesting approach, which we didn't include in our experiments (because of the results reported in [12]), is suggested by Durrieu et al. in [3], where a source-filter model is used for the vocals, and non-negative matrix factorisation (NMF) for the background.

3. EXPERIMENTS

In this Section, we discuss the outcome of three different strategies to utilise the results of the selected BSS algorithms.

3.1 Internal Data Set

For the first experiments, we use a set of 149 annotated rock songs by 149 different artists. All songs are recorded at a sampling rate of 22 kHz with 16-bit resolution and converted to mono. Background and vocal tracks are separately available to allow for a more complex evaluation of the results. Approximately 52% of the frames are annotated as vocal, and the amount of pure singing, i.e. without instrumental accompaniment, is negligible. This set is split into a 75 song train set, and a 74 song test set, approximately 5h each. It is challenging for BSS algorithms, because it contains lots of guitar soli, where singing voice characteristics are mimicked.

3.2 Feature Sets and Classifier

For the following experiments we choose the features from [9], which we refer to as *IC14* for the remainder of this paper. The IC14 feature vector comprises 116 attributes in total. This method was already compared to several others in [9], and turned out to deliver the best VD results in almost every testing scenario.

For new insights, we compare this feature set to the one used by W€eninger et al. in [22], henceforth referred to as *OS11*. This feature vector comprises 46 attributes. It was used along with a BLSTM-RNN classifier to achieve state-of-the-art performance for several singing voice related classification tasks, among them gender recognition and VD.

In our implementation, both feature sets are extracted with a fixed frequency of five observations per second (200 ms frames). Therefore, the units of audio to be classified are 200 ms frames. In the original implementation of OS11 in [22], the features were extracted beat-wise, hence using a variable framesize. As classifier, we choose the Random Forest (RF) as well as the Support Vector Machine (SVM) implementations of the Weka toolkit [7]. To be able to focus on the performance of feature set and classifier, no post-processing is applied.

3.3 Foreground Separation Evaluation

In this Section we present the results of the first strategy, where we extract the features (IC14 from [9], OS11 from [22]) from just the separated foreground audio signals.

The results are presented in Table 1, where we first see the performance of a model trained from the original audio, and tested with the original audio (row MIX). To simulate a perfect BSS, we additionally extract the features from the real vocal track (containing only vocals and silence) of the song, and test with the same model as before (row VOC). Clearly, the results improve by just using pure vocals as test data, e.g. from 83.7% to 91.6% accuracy for the IC14 feature set and the Random Forest (col. RF-accuracy-IC14).

For the upcoming results, we use the placeholder METHOD to refer to the four BSS algorithms {aREPET, FASST, KAM, RPCA} in general. For METHOD_{mix} classification, we always use the model trained from the mixed audio signals. For METHOD_{sep} classification, we use the model trained from the separated vocal signals to incorporate BSS characteristics.

The test data presented to the classifier is extracted from the separated vocals. As can be seen in Table 1, consistently and regardless of the feature set and classifier, both accuracy and F-measure improve when the model is trained with the separated vocals instead of the mixed audio data.

Nevertheless, there is quite some room for improvement, since all methods show a substantial performance decrease relative to testing with pure vocals (row VOC).

Compared to the results of training and testing with the mixed audio data (row MIX), only the aREPET_{sep} and RPCA_{sep} methods, where both training and testing is done with the separated foreground, yields slightly better results (e.g. for RF-accuracy-IC14: 83.7% vs. 84.1% and 84.5% respectively).

Interestingly, the feature set from [22] in combination with the SVM (col. SVM-accuracy-OS11) is only in the pure vocals scenario (row VOC) superior to the feature set from [9] (col. SVM-accuracy-IC14) (94.9% vs. 93.9% accuracy). It seems that the feature set from [22] is quite ca-

Internal Data Set (framesize=200ms)									
	RF				SVM				
	accuracy		F-measure		accuracy		F-measure		
	IC14	OS11	IC14	OS11	IC14	OS11	IC14	OS11	
MIX	.837	.795	.846	.814	.855	.807	.863	.819	
VOC	.916	.910	.920	.905	.939	.949	.943	.951	
aREPET _{mix}	.768	.756	.800	.781	.783	.742	.797	.789	
aREPET _{sep}	.841	.796	.850	.810	.861	.811	.866	.822	
FASST _{mix}	.732	.670	.682	.603	.751	.711	.756	.686	
FASST _{sep}	.826	.778	.835	.795	.845	.791	.854	.803	
KAM _{mix}	.752	.736	.773	.738	.631	.577	.728	.709	
KAM _{sep}	.826	.786	.835	.798	.849	.805	.855	.815	
RPCA _{mix}	.752	.691	.788	.763	.620	.563	.704	.703	
RPCA _{sep}	.845	.797	.851	.809	.861	.820	.867	.828	

Table 1. Results of Foreground Separation. F-measure relates to the class *vocal*. MIX: trained and tested with mixed audio; VOC: trained with mixed audio, tested with pure vocals. METHOD_{mix}: trained with mixed audio, tested with separated vocals; METHOD_{sep}: trained and tested with separated vocals. The columns IC14 and OS11 refer to the feature sets used in [9] and [22].

pable to model singing voice, but less robust to background noise.

Generally, comparing the performance of the classifiers, SVM delivers better results than the Random Forest. Regarding the feature set, IC14 seems to be the better choice. This can also be observed in the following experiments.

3.4 Foreground Concatenation Evaluation

Here, we concatenate the features extracted from the mix to the features extracted from the separated foreground into a single feature vector, hence doubling its size.

In Table 2 we can see that this strategy leads to better results regardless of BSS method, classifier and feature set. In order to assess the upper bound of this strategy, we include the results when using the real vocals also (row MIX+VOC), simulating perfect separation. Similar to Section 3.3, the results from utilising RPCA are the best, even though the absolute differences between the BSS methods are within 1 percentage point (ppt).

Compared to the previous strategy (see Section 3.3), the computational effort is much higher. This is especially true for training the SVM, due to the increased size of the feature vector. Therefore, we evaluate another strategy in the next Section, where the size of the feature vector stays the same instead of being doubled.

3.5 Foreground Enhancement Evaluation

In this Section we present the results of the third strategy to improve VD. In order to enhance the vocals (i.e. increase the SNR), we remix the separated foreground with the original signal. The mixes were made with different levels of the separated track, ranging from -6 dB to 6 dB in 3 dB steps. The results from 0 dB indicate, that the remix was done without any gain changes.

Training as well as testing was done by using the features extracted from the remixed signals. Again, we include the results when using the real vocals also. In Table 3 we can see that different gain changes for remixing do not make a big difference for the results, regardless of

Internal Data Set (framesize=200ms)									
	RF				SVM				
	accuracy		F-measure		accuracy		F-measure		
	IC14	OS11	IC14	OS11	IC14	OS11	IC14	OS11	
MIX	.837	.795	.846	.814	.855	.807	.863	.819	
MIX+VOC	.960	.985	.962	.986	.976	.984	.977	.985	
MIX+aREPET	.845	.800	.853	.817	.865	.825	.872	.834	
MIX+FASST	.842	.798	.850	.816	.863	.825	.871	.835	
MIX+KAM	.844	.800	.853	.815	.871	.830	.877	.839	
MIX+RPCA	.850	.806	.858	.822	.870	.833	.877	.841	

Table 2. Results of Foreground Concatenation. MIX: trained and tested with mixed audio. For training and testing of the methods aREPET, FASST, KAM, and RPCA, the classifier is given a double-sized vector containing the features from the mixed and the separated audio signal. MIX+VOC: concatenating features from the real vocals to simulate perfect separation.

the BSS method, classifier and feature set, except when using the real vocals (rows VOC). However, only the feature set from [9] allows for results at least as good as for the previous experiment in Section 3.4. Since those results are achieved without the additional computational burden due to a two-fold feature extraction, and the increased size of the feature vector, the enhancing-by-remixing strategy seems to be the best choice.

Again, RPCA based results are slightly better compared to the other source separation methods.

3.6 Final Method

Considering the results from the previous experiments, we choose the following setting for the upcoming experiments: For source separation, we choose the RPCA method, and we use the result to enhance the singing voice by remixing it with the original signal with an increased gain of 6 dB. The 116-attribute feature set IC14 as suggested in [9] is used, and fed to a SVM classifier with a radial basis function (RBF) kernel ($C = 2, \gamma = 0.35$). The remixed audios are used both for training and testing.

A very simple post-processing, where we use a median filter (order=5) for majority voting is also applied, which improves the results slightly from 87.3% to 87.8% accuracy.

3.7 Results on Public Data Sets

In this Section we compare the results of our suggested method as described in Section 3.6 to previously published results.

3.7.1 Jamendo

In [19], the authors presented results on a precisely defined split of the Jamendo data set, where the training set comprises 61 songs, and validation and test sets comprise 16 songs each. This allows for a fair comparison.

Table 4 lists the results reported by Lehner et al. in [9], compared with our new method. While the untouched output of the classifier (col. NEW) is on par with the (post-processed) baseline (col. LEH), the simple post-processing

	Internal Data Set (framesize=200ms)							
	RF				SVM			
	accuracy		F-measure		accuracy		F-measure	
IC14	OS11	IC14	OS11	IC14	OS11	IC14	OS11	
MIX	.837	.795	.846	.814	.855	.807	.863	.819
VOC -6dB	.880	.861	.886	.868	.907	.869	.911	.874
VOC -3dB	.895	.883	.900	.888	.922	.888	.925	.892
VOC 0dB	.909	.905	.914	.907	.937	.907	.939	.911
VOC 3dB	.923	.926	.926	.927	.949	.927	.951	.929
VOC 6dB	.937	.943	.940	.944	.960	.945	.961	.946
aREPET -6dB	.844	.792	.852	.809	.862	.807	.869	.818
aREPET -3dB	.843	.792	.852	.809	.863	.807	.871	.818
aREPET 0dB	.845	.794	.853	.811	.865	.809	.872	.820
aREPET 3dB	.845	.795	.854	.811	.866	.812	.873	.822
aREPET 6dB	.845	.799	.854	.813	.867	.813	.874	.823
FASST -6dB	.844	.795	.852	.812	.861	.805	.868	.817
FASST -3dB	.842	.796	.851	.813	.862	.807	.870	.819
FASST 0dB	.844	.799	.852	.816	.864	.809	.871	.820
FASST 3dB	.843	.799	.851	.816	.865	.811	.872	.822
FASST 6dB	.844	.799	.852	.815	.864	.811	.871	.822
KAM -6dB	.845	.801	.854	.816	.866	.815	.873	.825
KAM -3dB	.846	.803	.855	.817	.868	.818	.874	.827
KAM 0dB	.847	.804	.855	.817	.868	.819	.874	.828
KAM 3dB	.846	.803	.855	.816	.870	.820	.875	.829
KAM 6dB	.845	.803	.854	.816	.870	.821	.876	.829
RPCA -6dB	.847	.803	.855	.817	.868	.817	.874	.826
RPCA -3dB	.848	.805	.856	.819	.870	.818	.876	.827
RPCA 0dB	.851	.806	.858	.819	.871	.819	.877	.828
RPCA 3dB	.850	.807	.858	.819	.872	.820	.877	.829
RPCA 6dB	.850	.809	.858	.821	.873	.821	.878	.829

Table 3. Results of Foreground Enhancement. MIX: trained and tested with mixed audio. For training and testing of the methods aREPET, FASST, KAM, and RPCA, the classifier is given the features extracted from a signal, where the separated vocals are remixed with the original audio signal. VOC: using the real vocals instead of the separated.

	LEH	NEW	NEW+
accuracy	.882	.882	.896
recall	.862	.873	.892
precision	.880	.872	.884
F-measure	.871	.873	.888

Table 4. Jamendo corpus results. LEH: results reported in [9]. NEW: our new classifier (SVM) with RPCA based vocal enhancement. NEW+: incl. post-processing with median filter.

(see Section 3.6) helps to reach better results, with an accuracy of 89.6% (col. NEW+).

3.7.2 RWC

In [13], Mauch et al. report 87.2% accuracy with a 5-fold cross validation (CV) on a 102 song data set that is composed of 90 songs from the RWC music database [5], and 12 additional songs. Since we had just access to the 100 RWC songs, our results are only comparable to a certain extent. Therefore, we also include the (post-processed) results reported from Lehner et al. in [9] (col. LEH), where we could use exactly the same splits for the 5-fold CV.

In Table 5 we can see an improvement of 2.3 ppt accuracy by comparing LEH and NEW (87.5% vs. 89.8%), despite the lack of any post-processing. The post-processing (col. NEW+) did not improve the accuracy on this data set. However, the increased recall (0.928 vs. 0.939) could still be desired for certain use cases, even when it comes with reduced precision (0.905 vs. 0.898).

	MODE	MAUCH	MODE	LEH	NEW	NEW+
accuracy	.654	.872	.604	.875	.898	.898
recall	1.00	.921	1.00	.926	.928	.939
precision	.654	.887	.604	.875	.905	.898
F-measure	.791	.904	.753	.900	.917	.918

Table 5. Results on the RWC data set. MAUCH: results reported in [13]. NEW: our new classifier (SVM) with RPCA based vocal enhancement. NEW+: incl. post-processing with median filter. LEH and NEW were trained on the 100 RWC songs, MAUCH on 90 RWC + 12 additional (unknown) songs. MODE: baseline achievable by always predicting the majority class (*vocals*); MODE of classification accuracy thus tells the *percentage* of vocals in the data set.

4. IMPROVING BACKGROUND AND VOCAL ESTIMATES

In this Section, we discuss the results of BSS algorithms in more detail regarding the amount of non-vocal artifacts in the estimated vocals, and vocal artifacts in the estimated background.

All of the four presented BSS methods have one characteristic in common: they do not incorporate VD results. In [18], the authors even state that their REPET method does not require any explicit handling of singing voice segments. Although, by listening to the results of all presented BSS algorithms in this paper, we believe there is nevertheless room for improvement. Our internal data set contains a lot of instrumental soli, played by a guitarist. Considering the basic principle of e.g. the REPET method, it comes as no surprise that the estimates of the vocals have passages containing those solo instruments only, and no vocals whatsoever. This is especially troublesome for use cases like artist recognition. On the other hand, the estimates of the instrumental background often contain artifacts from the singing voice. This is problematic for tasks like automatic karaoke track creation.

In [2, 16], the vocal frames were already successfully used to improve the results of the source separation, but according to the annotated ground truth, and not to an automatic classification. Therefore, we investigate the impact of VD on the results of the BSS with respect to metrics traditionally used to evaluate BSS algorithms. Even though we consider only RPCA henceforth, the remaining three BSS methods show a very similar characteristic in that matter. Concerning the VD, we use the one improved by RPCA as described in Section 3.6.

We suggest a simple post-processing strategy to improve the estimates: Regarding the estimated vocals, we simply filter out (i.e. mute) the non-vocal frames. In other words, for the final estimates of the separated vocals, we decide whether to use the vocal estimates from RPCA or silence – according to our VD.

Figure 1 illustrates this principle, where we can see in the upper plot a time signal of vocals (dark) embedded in the mixture (bright). The lower plot shows the estimated

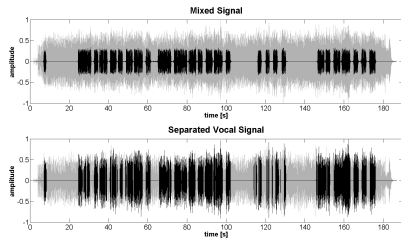


Figure 1. Example of RPCA separated singing voice. In the upper subplot we can see the mixed signal (bright) and the embedded singing voice (dark). In the lower subplot we can see the result from RPCA (bright) and the same result combined with singing voice detection (dark). Clearly, the latter approach is closer to the true singing voice.

vocals (bright) from the RPCA method, and the vocals after the VD based post-processing (dark). Obviously, the amount of non-vocal artifacts in the vocal estimate is reduced by applying this simple post-processing.

The same principle is applied in order to improve the estimates of the background. Here, we decide for the final estimates, whether to use the separated background or the original mix. This means, the separated background is only chosen, where the VD classifies the audio signal as vocal.

Nevertheless, it is not certain, if metrics traditionally used to evaluate BSS algorithms also reflect any improvement. A recall of vocals below an unknown – depending on the current situation – threshold would cause too much of the vocal estimates to be muted. At the same time, the estimated background would suffer from too much presence of vocals, since we would often wrongly opt for the original mixture instead of the separated background. Therefore, a thorough evaluation of the aforementioned post-processing is necessary in order to shed light on how useful it actually is.

4.1 Evaluation Metrics

In order to get meaningful evaluation results, we use the measurements proposed by Gribonval et al. in [6], where the overall estimation error is decomposed into *target distortion*, *interference*, and *artifacts*. Based on this components, the following energy ratios are defined: source Image to Spatial distortion Ratio (ISR), Source to Interference Ratio (SIR), and Source to Artifacts Ratio (SAR). Source to Distortion Ratio (SDR) is based on the three aforementioned measures, and serves as a global measure of distortion. For all metrics applies, that higher values indicate better performance.

Additionally, a set of measures that was proposed by Emiya et al. in [4] is used. Compared to the previously presented set, they better correlate with the perceived audio quality judged by human listeners. The overall distortion is also decomposed into the same three components, and based on them, the following measures are defined: Target-

related Perceptual Score (TPS), Interference-related Perceptual Score (IPS), and Artifacts-related Perceptual Score (APS). The Overall Perceptual Score (OPS) is based on the three aforementioned scores, and serves as a global measure of perceived audio quality. Similar to the aforementioned metrics, higher values indicate better performance. All measures were extracted with the PEASS toolkit [4].

4.2 Evaluation Results

In this Section, we present box plots of the evaluation results on our internal data set (see Section 3.1) regarding the VD based post-processing method, which we described in Section 4. Audio examples are available at <http://www.cp.jku.at/misc/ismir2015bss>.

4.2.1 Background

In Figure 2, we can see the evaluation results of the background, separated with RPCA. For each metric, we can see three results: raw RPCA output (*A*), RPCA output post-processed with VD (*B*), and RPCA output post-processed with ground-truth annotations (*C*). By adding the results from a ground-truth based post-processing, we assess the potential benefit of the suggested post-processing method, and how far away we are from this optimum.

Compared to the raw RPCA outputs *A*, the post-processed results *B* and *C* improve for all metrics, except IPS. The median of the global measure of distortion (SDR) improves by 2 dB for post-processing *B*, and 1.9 dB for post-processing *C* (*A*: 1.3 dB, *B*: 3.3 dB, *C*: 3.2 dB). This suggests, that our VD performs on par with using ground-truth.

The median of the global measure of perceived audio quality (OPS) improves by 6.5 points for post-processing *B*, and 8.3 points for post-processing *C* (*A*: 26.5, *B*: 33.0, *C*: 34.8). Even though the median OPS is approximately the same for post-processing *B* and *C*, we can see still room for improvement, since the distribution of the ground-truth based results *C* has a tendency towards higher values.

Interestingly, compared to the raw RPCA output *A*, the median of the IPS results drops for both post-processing methods. For the VD based results *B*, we assume, this is due to some missed vocals, where the original mix is chosen instead of the separated background. This causes the vocals to be moved back into the final background estimation, and deteriorates the result. For the ground-truth based results *C* we assume, this is due to the fact, that the vocal track that we use for evaluation, contains not complete silence, but rather some noise. But our final estimation of the vocals replaces non-vocal segments with silence.

Based on the results, we consider it useful to incorporate VD in order to yield better estimations of the background. This could be especially useful for generating karaoke tracks, where for the non-vocal segments the original mixture can be used, without any loss in quality due to BSS characteristics. Obviously, for songs with high vocal content, the impact will be rather small.

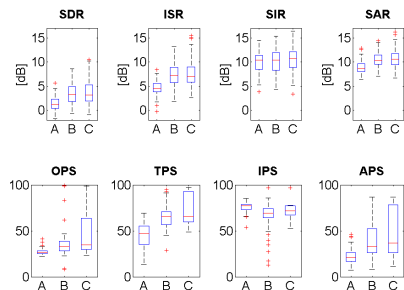


Figure 2. RPCA background estimation evaluation results. A: raw RPCA output; B: VD post-processed output; C: post-processed using ground truth. Higher values indicate better performance. In general, the performance increases for all metrics, except IPS. We assume, this is due to some missed vocals from our VD, where the original mix (incl. vocals) is chosen instead of the separated background.

4.2.2 Vocals

In Figure 3, we can see the evaluation results of the vocals, separated with RPCA. Compared to the raw RPCA outputs A, the median of SDR indicate better performance for the post-processed output B (-7.2 dB vs. -4.9 dB), and no improvement comparing post-processing B to C.

The impact of silencing all non-vocal segments for the final vocal estimates can be seen in the interference related SIR (A: -2.0 dB, B: 0.2 dB, C: 0.6 dB). The perceptually motivated IPS reveals this relationship even better, where we can see an improvement of 11.2 points for post-processing B and 12.5 points for post-processing C (A: 41.2, B: 52.4, C: 53.7).

The median of the OPS improves by 8.3 points for post-processing B, and 9.7 points for post-processing C (A: 10.9, B: 19.2, C: 20.6).

Similar to the background estimates, the results of the metrics indicate improvement, when VD based post-processing is applied. Especially for tasks like artist recognition it could be useful to only use the parts which are classified as vocals, even when some are missed by the VD.

5. CONCLUSION AND OUTLOOK

In this paper we first presented the outcome of three strategies of utilising different monaural BSS techniques to improve VD: foreground separation, foreground concatenation, and foreground enhancement. According to the results on an internal data set, foreground enhancement is the best strategy. The difference of the usefulness between the four techniques aREPET, FASST, KAM, and RPCA is relatively small, and the latter usually performs best. We compared the results achieved with the best approach on publicly available data sets, and could show an improvement of 2.3 ppt relative to the baseline, reaching an accuracy of 89.8% on the RWC data set. Compared to the same

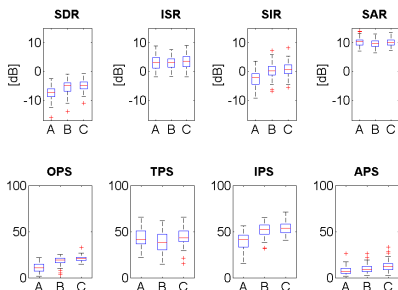


Figure 3. RPCA vocal estimation evaluation results. A: raw RPCA output; B: VD post-processed output; C: post-processed using ground truth. The global measures SDR and OPS indicate better performance for the post-processed output. The higher performance regarding interferences SIR and IPS are caused by the parts, that are muted, when our VD classifies them as non-vocal.

baseline, the results on the Jamendo data set have also improved by 1.4 ppt, with an accuracy of 89.6%. However, approximately half of the improvement is due to using a SVM instead of a Random Forest. Depending on the use case, the effort of employing a BSS might therefore not always be justified. Nevertheless, by adding the results obtained by using the real vocals, we could show that VD would principally benefit from better separation results.

Our second contribution addressed the issue, that all of the four separation techniques produce vocal estimates, where many segments contain only instrumental background, and no singing voice at all. We suggested to use the results of the VD to simply mute the non-singing parts. Regarding the vocal estimates, we could see an improvement of 2.3 dB SDR when applying this post-processing (-7.2 dB vs. -4.9 dB).

For the final background estimates, we suggested to use the original mixed audio signal, where the VD classifies the signal as non-vocal. Regarding the background estimates, we could see an improvement of 2.0 dB SDR when applying this post-processing (1.3 dB vs. 3.3 dB).

We think it is safe to conclude that VD based post-processing improves the results of BSS vocal and background estimates, although not by much regarding traditional evaluation metrics. However, in the context of vocalist recognition, it could be helpful to only use the classified vocal parts, especially when solo instruments like guitars cause the BSS algorithm to produce lots of non-vocal artifacts in the vocal estimates. As one of the next steps, we plan to investigate the usefulness of our approach in this topic.

6. ACKNOWLEDGEMENTS

This research is supported by the Austrian Science Fund (FWF) under grants TRP307-N23 and Z159.

7. REFERENCES

- [1] A. L. Berenzweig and D. P. W. Ellis. Locating singing voice segments within music signals. In *Workshop on the Applications of Signal Processing to Audio and Acoustics*, pages 119–122. IEEE, 2001.
- [2] T-S Chan, T-C Yeh, Z-C Fan, H-W Chen, L. Su, Y-H Yang, and R. Jang. Vocal activity informed singing voice separation with the ikala dataset. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*. IEEE, 2014.
- [3] J-L Durrieu, B. David, and G. Richard. A musically motivated mid-level representation for pitch estimation and musical audio source separation. *Journal of Selected Topics in Signal Processing*, 5(6):1180–1191, 2011.
- [4] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann. Subjective and objective quality assessment of audio source separation. *Transactions on Audio, Speech, and Language Processing*, 19(7):2046–2057, 2011.
- [5] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: Popular, classical, and jazz music databases. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, volume 2, pages 287–288, 2002.
- [6] R. Gribonval, L. Benaroya, E. Vincent, and C. Févotte. Proposals for performance measurement in source separation. In *4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003)*, pages 763–768, 2003.
- [7] M. Hall, F. Eibe, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The WEKA data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11(1):10–18, 2009.
- [8] P-S Huang, S. D. Chen, P. Smaragdīs, and M. Hasegawa-Johnson. Singing-voice separation from monaural recordings using robust principal component analysis. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012)*, pages 57–60. IEEE, 2012.
- [9] B. Lehner, G. Widmer, and R. Sonnleitner. On the reduction of false positives in singing voice detection. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2014)*, pages 7530–7534. IEEE, 2014.
- [10] A. Liutkus, D. Fitzgerald, Z. Rafii, B. Pardo, and L. Daudet. Kernel additive models for source separation. *Transactions on Signal Processing*, 62(16):4298–4310, 2014.
- [11] A. Liutkus, Z. Rafii, R. Badeau, B. Pardo, and G. Richard. Adaptive filtering for music/voice separation exploiting the repeating musical structure. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2012)*, pages 53–56. IEEE, 2012.
- [12] A. Liutkus, Z. Rafii, B. Pardo, D. Fitzgerald, L. Daudet, et al. Kernel spectrogram models for source separation. In *4th Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, pages 6–10. IEEE, 2014.
- [13] M. Mauch, H. Fujihara, K. Yoshii, and M. Goto. Timbre and Melody Features for the Recognition of Vocal Activity and Instrumental Solos in Polyphonic Music. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, pages 233–238, 2011.
- [14] T. L. Nwe, A. Shenoy, and Y. Wang. Singing voice detection in popular music. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 324–327. ACM, 2004.
- [15] A. Ozerov, E. Vincent, and F. Bimbot. A general flexible framework for the handling of prior information in audio source separation. *Transactions on Audio, Speech, and Language Processing*, 20(4):1118–1133, 2012.
- [16] Z. Rafii and B. Pardo. A simple music/voice separation method based on the extraction of the repeating musical structure. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2011)*, pages 221–224. IEEE, 2011.
- [17] Z. Rafii and B. Pardo. Music/voice separation using the similarity matrix. In *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR 2012)*, pages 583–588, 2012.
- [18] Z. Rafii and B. Pardo. Repeating pattern extraction technique (REPET): A simple method for music/voice separation. *Transactions on Audio, Speech, and Language Processing*, 21(1):73–84, 2013.
- [19] M. Ramona, G. Richard, and B. David. Vocal detection in music with support vector machines. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2008)*, pages 1885–1888. IEEE, 2008.
- [20] B. Schuller, A. Lehmann, F. Wening, F. Eyben, and G. Rigoll. Blind enhancement of the rhythmic and harmonic sections by nmf: Does it help. In *Proceedings of the International Conference on Acoustics (NAG/DAGA 2009)*, pages 361–364, 2009.
- [21] F. Wening, J-L. Durrieu, F. Eyben, G. Richard, and B. Schuller. Combining monaural source separation with long short-term memory for increased robustness in vocalist gender recognition. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2011)*, pages 2196–2199. IEEE, 2011.
- [22] F. Wening, M. Wöllmer, and B. Schuller. Automatic assessment of singer traits in popular music: Gender, age, height and race. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, pages 37–42, 2011.

DETECTION OF COMMON MISTAKES IN NOVICE VIOLIN PLAYING

Yin-Jyun Luo^{1,2}

Li Su²

Yi-Hsuan Yang²

Tai-Shih Chi¹

¹Department of Electrical and Computer Engineering, National Chiao Tung University, Taiwan

²Research Center for Information Technology Innovation, Academia Sinica, Taiwan
fredom.smt02g@nctu.edu.tw, lisu@citi.sinica.edu.tw,
yang@citi.sinica.edu.tw, tschi@mail.nctu.edu.tw

ABSTRACT

Analyzing and modeling playing mistakes are essential parts of computer-aided education tools in learning musical instruments. In this paper, we present a system for identifying four types of mistakes commonly made by novice violin players. We construct a new dataset comprising of 981 legato notes played by 10 players across different skill levels, and have violin experts annotate all possible mistakes associated with each note by listening to the recordings. Five feature representations are generated from the same feature set with different scales, including two note-level representations and three segment-level representations of the onset, sustain and offset, and are tested for automatically identifying playing mistakes. Performance is evaluated under the framework of using the Fisher score for feature selection and the support vector machine for classification. Results show that the F-measures using different feature representations can vary up to 20% for two types of playing mistakes. It demonstrates the different sensitivities of each feature representation to different mistakes. Moreover, our results suggest that the standard audio features such as MFCCs are not good enough and more advanced feature design may be needed.

1. INTRODUCTION

With advances in music technology, the development of computer-aided music learning and automatic scoring systems has attracted wide attention. Such systems provide self-learning experiences to users through computer-aided platforms. Despite numerous efforts have been made, however, the performance of current systems still leaves plenty of space for improvement. A review of the music learning system and the main challenge can be found in [1].

For a novice player, three common basic aspects, intonation, rhythm and timbre, are often used to evaluate his/her performance [2]. Intonation refers to the pitch of the tone, rhythm specifies the duration of the tone, and timbre characterizes the overall quality of the tone. Con-

ventionally, a novice player uses a tuner for correcting the intonation and a metronome for following the rhythm during the practice. In traditional music education, there is no hardware device capable of automatically evaluating the timbre quality.

Up to date, most of the computer-aided music learning systems also focus on intonation and rhythm only [1]. These studies mainly coped with learning intonation and rhythm in music in the context of automatic music transcription (AMT). For example, the pitch played by the violin learner was automatically detected and visually presented to evaluate the pitch intonation [3]. A fusion of audio and video cues improved the onset detection of non-percussive instruments, such as violin, and thereby enhanced the performance of AMT [4]. Automatic singing quality assessment is achieved by measuring the dissimilarity between singing voices of beginners and of trained singers [5]. Besides intonation and rhythm, timbre plays an essential role in identifying the skill (or proficiency) level of a player but has not attracted much attention in computer-aided music learning platforms. Some timbre-related research studies considered instrumental expression to recognize the techniques in playing musical notes by violin [6] and by electric bass guitar [7], respectively. Other studies aimed to evaluate the played notes, for example, using spectral parameters from long tones to evaluate the technical level of saxophone players [8]. Recently, a hierarchical approach combining deterministic signal processing and deep learning was employed to identify different common mistakes made by novice flute players [9]. Machine learning techniques were also adopted to distinguish good trumpet tones from bad ones [10]. The first attempt to detect bad violin playing in [11] is the most relevant work to the proposed study. One of the two tasks conducted in [11] classifies violin tones into binary clusters, i.e., good or bad, using k-nearest neighborhood algorithm. The other task examined the prominent feature sets for detecting individual playing mistakes. Similarly, in this paper, we explore the capability of timbre in detecting playing mistakes produced by novice violin players during practice. However, since the dataset and the algorithm codes in [11] are not publicly available, it is difficult to compare our approach with the approach in [11].



© Yin-Jyun Luo, Li Su, Yi-Hsuan Yang and Tai-Shih Chi. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Yin-Jyun Luo, Li Su, Yi-Hsuan Yang and Tai-Shih Chi. "Detection of Common Mistakes in Novice Violin Playing", 16th International Society for Music Information Retrieval Conference, 2015.

The first contribution of this paper is to build and release a new dataset¹ for such a research problem. To be as realistic as possible, we recorded four successive legato notes, which require smooth, round and continuous flow of tones [12], as a unit and then trimmed it into individual notes rather than simply recording single note at a time as in [11]. The resulting dataset comprises of 981 individual legato notes played by several players across different skill levels. The playing mistakes associated with each note were annotated by violin experts using the following four pre-defined classes: *scratching bow*, *crooked bow*, *bouncing bow*, and *inappropriate arm height*. More details of the annotations and dataset are elaborated in Sections 2 and 3, respectively.

The second contribution of this paper is to evaluate a number of features capturing the acoustic characteristics of different segments of a musical note for the task of automatic playing mistake classification. A set of spectral features is extracted from either the whole note or the segment of onset, sustain and offset, partitioned using the output of an optical sensor installed on the violin. The approach leads to five different feature representations: *Note*, *Onset*, *Sustain*, *Offset* and *Cascade*. They refer to features extracted from the whole note, from the corresponding segments only and the concatenation of segment-level features, respectively. More details about the partitioning method and feature extraction can be found in Sections 3 and 4, respectively.

In our approach, the Fisher score is used for feature selection and the support vector machine (SVM) is used for classification. Feature selection is done as a preprocessing step of classification. The performance of classification is assessed in terms of the F-measure. Experimental details are presented in Section 5. Exploration of insights to link specific feature representations to playing mistakes is presented in Section 6, before we conclude the paper in Section 7.

2. VIOLIN PLAYING MISTAKES

We defined four common playing mistakes made by violin novices. These mistakes are mainly related to the bow arm and the bow hand which dominantly control violin timbre for novice players and cause most of the trouble for violinists [2].

2.1 Scratching Bow (SB)

The pressure of the bow applied on the string can either come from the weight of the bow, arm and hand, from controlled muscular action, or from a combination of these factors [2]. Excessive bowing pressure without enough bowing speed to complement with can hinder the vibrations of the string and produce coarse sound with inferior quality. Without the support of bowing speed, extreme

pressure of the bow on the string results in sound with scratching effect.

2.2 Crooked Bow (CB)

Drawing a straight bow from the frog to the tip is the foundation of the bowing technique [2]. If the bow is crooked, not parallel to the bridge, the sound quality will vary due to change of the contact position of the bow on the string. Severe inclination even causes sudden displacement of the bow from the bridge and produces sound with skating effect.

2.3 Bouncing Bow (BB)

Lack of muscular control of either the bow arm or the bow grip reduces strength to the bow. It might prevent the bow from properly laying on the string, thereby the bow bounces naturally due to its elasticity.

2.4 Inappropriate Arm Height (IAH)

Appropriate tilt of the arm relative to the bow is required in order to play on each string without touching the other strings. With inconsistent height or tilt of the arm when drawing the bow across the string, pitch produced by adjacent string might be heard.

3. DATASET

All notes in the dataset were played by ten players across different skill levels using the same violin in a semi-anechoic chamber. Four players are relatively more experienced in violin or similar string instruments such as cello, while the other six players have learned to play violin for less than one month. Each player was asked to play four successive notes as a clip at the speed of 60 beat-per-minute (BPM). Each clip was directed to start with down-bow and end with up-bow. In total, 26 clips containing 104 legato notes were played by each player. This style of successive playing is more similar to actual practicing than the style of playing an individual note at once. In our recordings, analysis of transition between notes is also feasible though we leave it as future work. We limit the study to consider legato notes only because legato is the essence of all cantabile playing [12] and one can hardly master other advanced techniques before playing it well.

Segmentation between notes and within each note was achieved using a photo resistor and four rings of surface-mounted light-emitting diodes (SMD LEDs) installed respectively underneath the violin bridge and on the bow stick. Two of the four rings were installed at the positions close to the frog and the tip on the bow stick, while the other two were placed at both ends of the middle of the bow. Segmenting a violin note can benefit the analysis, as the time domain signal varies in characteristic over a bow draw. The purpose of installing the optical sensor was to segment the time domain signal in a more direct way rather than the approach in [13]. When a legato note was played, the optical sensor was capable of marking the time instants, at which those ring-located

¹ The audio clips and annotations of playing mistakes can be found in <http://perception.cm.nctu.edu.tw/sound-demo/>.

positions of the bow stick passed through the violin bridge, without influencing playing. As our main purpose was to simply divide the bow draw into three segments, we can tolerate the small accuracy errors of the sensors on the longitudinal bow position [13].

Based on the marked time instants, we divided each clip into four individual notes and segmented each note into three different segments, i.e., onset, sustain and offset, as playing mistakes can occur at any instant of the drawing. The two ends of each clip, the start of the first note and the end of the fourth note, were manually determined by an energy threshold. The edges between successive notes and within each individual note were automatically defined by the marked time instants. At the end, we collected 981 notes in total and the corresponding segments after discarding notes containing accidentally made distinct noise during the recording.

We employed the hardware-assisted approach instead of automatic approaches proposed in the literature [14, 15] because automatic approaches usually segment an individual musical note according to the temporal evolution of amplitude envelope and spectral centroid [14, 15]. Since we are dealing with notes produced by the violin novices, the algorithms developed for well-played musical notes are not applicable in our case. For instance, simply dividing the note into three equal-duration segments would not produce same results as our hardware-assisted approach since novice players cannot draw the bow with a constant speed. Therefore, without the assistance of the optical sensors, automatic segmentation of violin notes performed by novice players should be a difficult task, which is beyond the scope of this paper and deserves further research in the future.

The notes were then annotated by violin experts using the four pre-defined mistakes. Note that a single note could possess multiple playing mistakes. Fig. 1 shows the duration distributions of notes and the corresponding segments. One can observe that although players were asked to play each note at the speed of 60 BPM, beginners, especially those who lack of musical background, weren't necessarily able to perform accurately. Table 1 summarizes the numbers of instances of the playing mistakes in the first row. Dividing the first row by the total number of the collected notes gives the percentages in the second row.

4. METHOD

4.1 Preprocessing

All of the notes in the dataset were resampled to 44.1 kHz and saved in the mono-channel WAV format. Before feature extraction, each time domain signal was first normalized to zero mean and unit variance and then divided into three segments as described in Section 3 for further analysis.

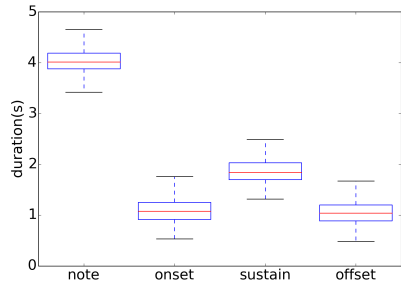


Figure 1. Duration distributions of 981 notes (the first column), the corresponding onset, sustain and offset segments (the second to the last column).

	SB	CB	BB	IAH
Numbers	265	133	154	53
Percentage	27.0%	13.6%	15.7%	5.4%

Table 2. The number of instances of each mistake and the corresponding percentage.

4.2 Feature Extraction

A set of 30 frame-level spectral features, including high frequency content (HFC) [16], 13 Mel-frequency cepstral coefficients (MFCCs), spectral centroid, spectral crest, spectral flatness, spectral flux, spectral roll-off, descriptors of spectral distribution (i.e., spectral variance, skewness and kurtosis), tristimulus [17], odd-to-even harmonic energy ratio (OER) [18], the estimated pitch, zero crossing rate and the instant power, were extracted from either the waveform or the spectrum using the ESSENTIA open-source library (version 2.0.1) [19]. The feature extraction was performed in each Hanning-windowed frame with the frame duration of 46 ms and the frame shift of 50%. These features are capable of characterizing timbre and regularly employed in audio signal processing applications [20]. The six temporal functionals, including mean, variance, skewness, kurtosis, mean and variance of the derivative, of all the frame-level features were derived to generate clip- or segment-level features. The outcome of the feature extraction stage is a feature vector of 180 dimensions.

The feature extraction process was done on different segments of notes resulting in five feature representations: *Note*, *Onset*, *Sustain*, *Offset* and *Cascade*. The *Note* representation was extracted from each intact note while the *Onset*, *Sustain* and *Offset* representations were extracted from corresponding segments of each note. These four representations consist of feature vectors of 180 dimensions. The *Cascade* representation was produced by concatenating the *Onset*, *Sustain* and *Offset* representations

to give a 540-dimensional feature vector for each note. All feature representations were derived from 981 recorded notes and used for the task of playing mistake classification.

4.3 Feature Selection and Classification

The Fisher score was considered for selecting prominent features in a pre-processing step prior to classification to reduce amounts of computation [21]. It is defined as [22]

$$F(i) \equiv \frac{(\bar{x}_i^{(+)} - \bar{x}_i)^2 + (\bar{x}_i^{(-)} - \bar{x}_i)^2}{\frac{1}{n^{(+)}-1} \sum_{k=1}^{n^{(+)}} (\bar{x}_{k,i}^{(+)} - \bar{x}_i^{(+)})^2 + \frac{1}{n^{(-)}-1} \sum_{k=1}^{n^{(-)}} (\bar{x}_{k,i}^{(-)} - \bar{x}_i^{(-)})^2},$$

where $n^{(+)}$ and $n^{(-)}$ are the numbers of positive and negative instances, respectively; \bar{x}_i , $\bar{x}_i^{(+)}$, $\bar{x}_i^{(-)}$ are the averages of the i th feature over the whole, positive, and negative instances, respectively; $\bar{x}_{k,i}^{(+)}$ is the i th feature of the k th positive instance, and $\bar{x}_{k,i}^{(-)}$ is the i th feature of the k th negative instance.

We followed the framework in [22] which selects features with high Fisher scores and uses the support vector machine (SVM), implemented by LIBSVM [23], for classification. The performance was evaluated in terms of the averaged F-measure, which is the harmonic mean of precision and recall, for each mistake using each feature representation with 100 repetitions of stratified five-fold cross-validation (CV).

5. EXPERIMENTS

The goal of the experiments is to investigate the capability of used features to detect playing mistakes and bridge the relation between playing mistakes and feature representations from different segments of notes.

Detection experiments were carried out through all feature representations after completing feature extraction. Following the procedures in [22], we first adopted a nested stratified five-fold CV to find the best percentage threshold to retain features based on Fisher scores, and then used the selected features for grid searching the optimized hyper-parameters C and γ , from the choices of $\{10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}\}$ and $\{1, 10, 100, 1000\}$ respectively, of the radial-basis function (RBF) kernel based SVM. Finally, the selected threshold and the hyper-parameters were fed into another stratified five-fold CV. The overall performance was evaluated by averaging the F-measures of 100 repetitions of the final CV. Note that the above experiments were conducted for each feature representation and for each mistake. In other words, we trained M binary SVMs on each feature representation, where M is the number of types of playing mistakes.

To further have subset analysis, the experiments were conducted using three sets of data: *All*, *Down-bow*, and *Up-bow*, which respectively refer to the full data set of

981 notes, the set of 480 notes played with down-bow and the set of 481 notes played with up-bow. Moreover, we performed the same experiment on the 570 notes recorded by the six beginners who have played violin less than one month. Experiment results on these subset data and related discussions will be given in the next section.

6. RESULTS

The averaged F-measure using each feature representation for identifying each playing mistake in the *All* dataset is shown in Fig. 2. One can see that *Cascade* performs slightly better than *Note* in terms of the F-measure across all the mistakes, which is verified by the two-tailed t -test ($p < 0.01$). It is probably because *Cascade* contains more detailed information of each individual segment. Except for the BB mistake, *Cascade* performs better than each of its constituents, i.e., *Onset*, *Sustain* and *Offset*. Note that the F-measures of the playing mistakes by the random guess would be 35.0%, 21.3%, 23.7% and 9.7%, respectively, equivalent to the prior probabilities $p(m)$ of the mistake m as shown in the second row of Table 1 divided by $p(m) + 0.5$. It is because we preserved the prior distribution of the dataset in all partitions during the stratified five-fold CV procedures for each playing mistake. For comparison, we show in Fig 3 the performance of using the original 180 features without feature selection. Similar results between Figs. 2 and 3 suggest that the selected features sufficiently capture information embedded in the original 180 features for our experiments.

To explore more connections between playing mistakes and feature representations, one can re-arrange the F-measures of *Onset*, *Sustain* and *Offset* against mistakes as in Table 2. Results in Table 2 show that *Onset* has advantage in detecting SB over the others. It means that the onset segment is more sensitive for detecting SB, which somehow implies that the 10 players tended to have excessive bow pressure at the beginning of the bow draw. In contrast, *Sustain* surpasses the others in both CB and BB by up to 8% and 20%, respectively, which suggests CB and BB have higher chance to emerge during the middle of a drawing bow. Lastly, *Offset* dominates the IAH mistake. Such ‘‘favor’’ of a specific playing mistake in a particular segment of a note reveals the tendency of players to make that mistake at certain moment of a bow draw. This kind of information is helpful to novice players during their practice.

As shown in Table 2, SB and BB are prone to happen in the onset segment and sustain segment, respectively. Furthermore, it is commented by violin experts that such ‘‘favor’’ of SB and BB would be even more obvious in down-bow notes based on their teaching experiences. Figs. 4 and 5 compare the F-measures between the *Up-bow* and *Down-bow* subsets for the SB and BB mistakes, respectively. Obviously, these two figures indicate the down-bow notes are more associated with the mistakes than the up-bow notes, which is consistent with experiences of the violin experts.

Moreover, Fig. 6 shows the results on notes only played by the six beginners. It shows better overall performance

than results in Fig. 2, which suggests notes played by the beginners reveal more obvious characteristics of mistakes than the ones played by experienced players. In other words, the adopted features might be incapable of capturing slight mistakes made by experienced players.

The inferior performance in classifying the IAH mistake, as shown in Figs. 2 and 6, might result from the severe imbalance of the dataset. In addition, pitch-related features are overwhelmed by timbre-related features in our adopted feature set. If more pitch features are considered, it is possible to further improve the performance for IAH detection, since it is about the mistake of playing undesired pitch.

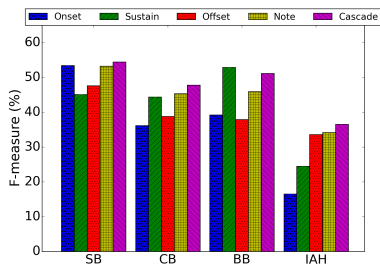


Figure 2. Averaged F-measures of playing mistake classification on all recorded notes using different feature representations.

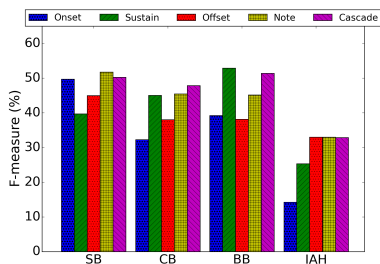


Figure 3. Average F-measures of playing mistake classification on all recorded notes using different feature representations from the original 180 features.

	SB	CB	BB	IAH
<i>Onset</i>	53.4	36.1	39.2	16.5
<i>Sustain</i>	45.0	44.3	52.9	24.4
<i>Offset</i>	47.6	38.8	37.9	33.5

Table 2. Averaged F-measures (in %) of *Onset*, *Sustain* and *Offset*. The feature representation with the highest F-measure for each mistake is highlighted.

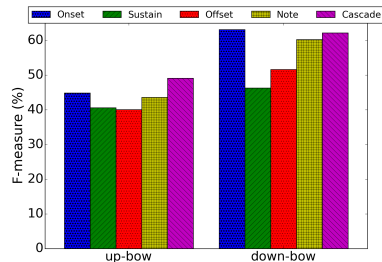


Figure 4. Averaged F-measures of the playing mistake 'scratching bow' (SB) using different feature representations within the *up-bow* and *down-bow* subsets.

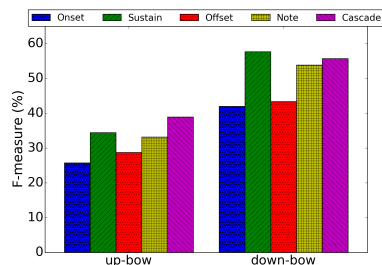


Figure 5. Averaged F-measures of the playing mistake 'bouncing bow' (BB) using different feature representations within the *up-bow* and *down-bow* subsets.

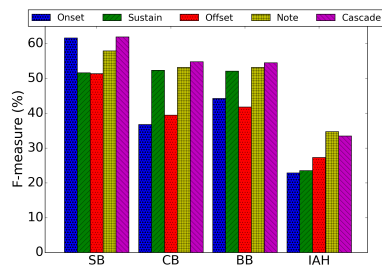


Figure 6. Averaged F-measures of playing mistake classification on notes played by beginners using different feature representations.

7. CONCLUSION AND FUTURE WORK

In this study, we first recorded a new dataset of violin legato notes played by novice players. Then we defined four common playing mistakes mainly made by bow arm

and performed automatic playing mistake classification using spectral and temporal features extracted from different segments of the notes.

Our evaluation on different feature representations suggests concatenation of segment-level features provides more information than the note-level features in identifying playing mistakes. Furthermore, by exploring connections between playing mistakes and feature representations, we found SB, CB, BB, and IAH mistakes are prone to happen in the onset, sustain, sustain, and offset segments, respectively. These findings would serve pedagogical purpose and benefit novice violin players. Our future work will focus on improving the overall classification performance by enriching the dataset and seeking more relevant features, using either feature design or feature learning techniques [24, 25].

8. ACKNOWLEDGEMENTS

This research is supported by the National Science Council, Taiwan under Grant No NSC 102-2220-E-009-049, the Biomedical Electronics Translational Research Center, NCTU, and the Academia Sinica Career Development Award.

9. REFERENCE

- [1] C. Dittmar, E. Cano, J. Abeßer and S. Grollmisch: "Music information retrieval meets music education," *Multimodal Music Processing*, Dagstuhl Follow-Ups M. Müller, M. Goto and M. Schedl, Eds., vol. 3, pp. 95–120, 2012.
- [2] I. Galamian: *Principals of Violin Playing and Teaching*, London, Prentice Hall, 1985.
- [3] J. Wang, S. Wang, W. Chen, K. Chang and H. Chen: "Real-Time Pitch Training System for Violin Learners," *Proc. IEEE Int. Conf. Multimedia and Expo Workshops*, pp. 163–168, 2012.
- [4] B. Zhang and Y. Wang: *Automatic music transcription using audio-visual fusion for violin practice in home environment*, Tech. Report TRA7/09, School of Computing, National University of Singapore, 2009.
- [5] E. Molina *et al.*: "Fundamental frequency alignment vs. note-based melodic similarity for singing voice assessment," *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 744–748, 2013.
- [6] I. Barbancho, C. Bandera, A.M. Barbancho and L. J. Tarón: "Transcription and Expressiveness Detection System for Violin Music," *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 189–192, 2013.
- [7] J. Abeßer *et al.*: "Feature-based extraction of plucking and expression styles of the electric bass guitar," *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, pp. 2290–2293, 2010.
- [8] M. Robine and M. Lagrange: "Evaluation of the Technical Level of Saxophone Performers by Considering the Evolution of Spectral Parameters of the Sound," *Proc. Int. Society for Music Information Retrieval Conference*, pp. 79–84, 2006.
- [9] Y. Han and K. Lee: "Hierarchical approach to detect common mistakes of beginner flute players," *Proc. Int. Society for Music Information Retrieval Conference*, pp. 77–82, 2014.
- [10] T. Knight, F. Upham, and I. Fujinaga: "The potential for automatic assessment of trumpet tone quality," *Proc. Int. Society for Music Information Retrieval Conference*, pp. 573–578, 2011.
- [11] J. Charles: *Playing Technique and Violin Timbre: Detecting Bad Playing*, Ph.D. dissertation, Dublin Institute of Technology, 2010.
- [12] L. Auer: *Violin Playing As I Teach It*, Dover Publications Inc., New York, 1980.
- [13] T. Grosshauser, and T. Gerhard: "Optical bow position, speed and contact point detection," *Proc. ACM Int. Conf. Pervasive and Ubiquitous Computing Adjunct Publication*, 2013.
- [14] M. Hajda: "A New Model for Segmenting the Envelope of Musical Signals: The Relative Salience of Steady State Versus Attack, Revisited," *Audio Eng. Soc*, paper No. 4391, 1996.
- [15] M. Caetano, J.J. Burred and X. Rodet: "Automatic Segmentation of the Temporal Evolution of Isolated Acoustic Musical Instrument Sounds Using Spectro-Temporal Cues," *Proc. Int. Conf. Digital Audio Effects*, 2010.
- [16] P. Masri and A. Bateman: "Improved modelling of attack transients in music analysis-resynthesis," *Proc. Int. Computer Music Conference*, pp. 100–103, 1996.
- [17] G. Peeters: *A large set of audio features for sound description (similarity and classification) in the CUIDADO project*, CUIDADO I.S.T. Project Report, 2004.
- [18] K. D. Martin and Y. E. Kim: "Musical instrument identification: A pattern-recognition approach," *The Journal of the Acoustical Society of America*, vol. 104, no. 3, pp. 1768–1768, 1998.
- [19] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera and O. Mayor: "ESSENTIA: an audio analysis library for music information retrieval,"

Proc. Int. Society for Music Information Retrieval Conference, pp. 493-498, 2013.

- [20] M. Müller et al: "Signal processing for music analysis," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 6, pp. 1088-1110, 2011.
- [21] I. Guyon and A. Elisseeff: "An introduction to variable and feature selection," *Journal of Machine Learning Research*, vol. 3, pp. 1157-1182, 2003.
- [22] Y.-W. Chen and C.-J. Lin: "Combining SVMs with various feature selection strategies," *Feature extraction foundations and applications*, pp. 315-324, 2006.
- [23] C.-C. Chang and C.-J. Lin: "LIBSVM : a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, 2:27:1--27:27, 2011.
- [24] P.-C. Li, L. Su, Y.-H. Yang and A. W. Y. Su: "Analysis of expressive musical terms in violin using score-informed and expression-based audio features," *Proc. Int. Society for Music Information Retrieval Conf.*, 2015.
- [25] L. Su, L.-F. Yu and Y.-H. Yang: "Sparse cepstral and phase codes for guitar playing technique classification," *Proc. Int. Society for Music Information Retrieval Conf.*, pp. 9-14, 2014.

PROBABILISTIC MODULAR BASS VOICE LEADING IN MELODIC HARMONISATION

Dimos Makris
Department of Informatics,
Ionian University,
Corfu, Greece
c12makr@ionio.gr

Maximos Kaliakatsos-Papakostas
School of Music Studies,
Aristotle University of
Thessaloniki, Greece
maxk@mus.auth.gr

Emilios Cambouropoulos
School of Music Studies,
Aristotle University of
Thessaloniki, Greece
emilios@mus.auth.gr

ABSTRACT

Probabilistic methodologies provide successful tools for automated music composition, such as melodic harmonisation, since they capture statistical rules of the music idioms they are trained with. Proposed methodologies focus either on specific aspects of harmony (e.g., generating abstract chord symbols) or incorporate the determination of many harmonic characteristics in a single probabilistic generative scheme. This paper addresses the problem of assigning voice leading focussing on the bass voice, i.e., the realisation of the actual bass pitches of an abstract chord sequence, under the scope of a modular melodic harmonisation system where different aspects of the generative process are arranged by different modules. The proposed technique defines the motion of the bass voice according to several statistical aspects: melody voice contour, previous bass line motion, bass-to-melody distances and statistics regarding inversions and note doublings in chords. The aforementioned aspects of voicing are modular, i.e., each criterion is defined by independent statistical learning tools. Experimental results on diverse music idioms indicate that the proposed methodology captures efficiently the voice layout characteristics of each idiom, whilst additional analyses on separate statistically trained modules reveal distinctive aspects of each idiom. The proposed system is designed to be flexible and adaptable (for instance, for the generation of novel blended melodic harmonisations).

1. INTRODUCTION

In melodic harmonisation systems harmony is expressed as a sequence of chords, but an important aspect is also the relative placement of the notes that comprise chord sequence, which is known as the *voice leading* problem. As in many aspects of harmony, in voice leading there are certain sets of diverse conventions for different music *idioms*

that need to be taken under consideration. Such rules have been hand-coded by music experts for the development of rule-based melodic harmonisation systems (see [15] for a review of such methods). Similarly, such hand-coded rules have been utilised as fitness criteria for evolutionary systems (see [4, 18] among others). However, the specification of rules that are embedded within these systems are very complex with many variations and exceptions. Additionally, the formalisation of such rules has not yet been approached for musical idioms that have not hitherto been thoroughly studied. Most of the works so far, have focused on either finding a satisfactory chord sequence for a given melody (performed by the soprano voice), or on completing the remaining three voices that constitute the harmony for a given melodic or bass line (known as the “four-part harmony” task) [5, 14, 18, 24]. Experimental evaluation of methodologies that utilise statistical machine learning techniques demonstrated that an efficient way to harmonise a melody is to add the bass line first [22]. To this end, the motivation behind the work presented in the paper at hand is further enforced by the findings in the aforementioned paper.

This study, is based on the following underlying melodic harmonisation strategy: 1) analyse a give melody in terms of segmentation, scale/pitch hierarchy, harmonic/embellishment notes, harmonic rhythm (this can be achieved automatically or, at this stage, manually), 2) assign abstract chords to the given melody from learned first-order chord transition tables, 3) select concrete pitches from abstract chords for the bass-line based on learned melody-to-bass-line movement (discussed in this paper), 4) select concrete pitches for inner voices (steady or varied number of notes per chord). This scheme would seem to be adequate for a large body of non-monophonic music, but not all. For instance, even the mere concept of chords (with inversions) is rather controversial in European music before the mid-eighteenth century and in other traditional polyphonic musics; more so, the idea of melody with chords and functional bass line is untenable in such music.

However, as the aim of this project is not individual fully-fleshed harmonic models of different idioms, but rather a general-as-possible method to ‘extract’ basic components of harmonic content in various harmonic textures, it is possible to employ the above strategy in any non-monophonic texture. It is known that outer voices tend to stand out per-



© Dimos Makris, Maximos Kaliakatsos-Papakostas, Emilios Cambouropoulos.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Dimos Makris, Maximos Kaliakatsos-Papakostas, Emilios Cambouropoulos. “Probabilistic modular bass voice leading in melodic harmonisation”, 16th International Society for Music Information Retrieval Conference, 2015.

ceptually (e.g. in [6]); additionally, note simultaneities can be encoded in a more abstract manner (e.g., GCT representation). Employing a computational methodology based on such generic concepts, can enable the construction of a ‘generic’ melodic harmoniser that can use harmonic components from various idioms, without claiming to emulate the idioms themselves.

This paper proposes a modular methodology for determining the bass voice leading, to be integrated in a melodic harmonisation system under development. The effectiveness of the proposed methodology that performs bass voice leading according to statistics describing the overall voicing layout (i.e. arrangement of pitches) of given chord sequences in the General Chord Type (GCT) [2] representation is examined. This methodology is extending the bass voice leading scheme presented in [12], by harnessing voicing layout information through additional voicing layout statistical, independently trained, *modules* concerning the chords that constitute the harmonisation. Those characteristics include distributions on the distance between the bass and the melody voice and statistics regarding the inversions and doublings of the chords in the given chord sequence. By training these modules on multiple diverse idioms, a deeper study is pursued within the context of the COINVENT project [20], which examines the development of a computationally feasible model for conceptual blending. Thereby, blending different modules from different idioms will expectedly lead to harmonisations with blended characteristics.

2. PROBABILISTIC MODULAR BASS VOICE LEADING

Given the fact that a melody is available in systems that perform melodic harmonisation, the methodology presented in [12] derives information from the melody voice in order to calculate the most probable movement for the bass voice, named as the *bass voice leading* (BVL). This approach, in combination with information regarding the *voice layout* (Section 2.2), is incorporated into a *larger modular probabilistic framework*. In the integrated modular melodic harmonisation system under development, the selection of chords (in GCT form [2]) is performed by another probabilistic module [10] not discussed in this paper. Therefore, the herein discussed modules have been developed to provide indications about possible movement of the bass as well as to define specific notes for the bass voice, providing a first step to complete information regarding specific voices from the chords provided by the chord selection module.

To this end, both the bass and the melody voice steps are represented by abstract notions that describe general quantitative information on pitch direction. In [12] several scenarios for voice contour refinement were examined, providing different levels of accuracy for describing the bass motion in different datasets. In the paper at hand, the selected methodology is the one with the greatest level of detail, i.e. the scenario where the melody and bass note changes are divided in seven steps, as exhibited

in Table 1. While different range schemes could have been selected, the rationale behind the utilised one is that the perfect fourth is considered as a small leap and the perfect fifth as a big leap.

refinement	short name	range (semitones)
steady voice	st_v	$x = 0$
step up	s_up	$1 \leq x \leq 2$
step down	s_down	$-2 \leq x \leq -1$
small leap up	sl_up	$3 \leq x \leq 5$
small leap down	sl_down	$-5 \leq x \leq -3$
big leap up	bl_up	$5 < x$
big leap down	bl_down	$x < -5$

Table 1. The pitch step and direction refinement scale considered for the development of the utilised bass voice leading system.

2.1 The hidden Markov model module

The primary module for defining bass motion functions under the first order Markov assumption in combination with the fact that it depends on the piece’s melody. To this end, the next step of the bass voice contour (bass direction descriptor) is dependent on the previous one and on the current melody contour (melody direction descriptor). This assumption, based on the fact that a probabilistic framework is required for the harmonisation system, motivates the utilisation of the *hidden Markov model* (HMM) methodology. According to the HMM methodology, a sequence of observed elements (melody direction descriptor) is given and a sequence of (hidden) states (bass direction descriptor) is produced as output. The ‘order’ of the HMM utilised in the presented work, i.e. how many previous steps are considered to define the current, is 1. In melodic harmonisation literature different orders have been examined, e.g. [19], where it is shown that order 1 might not be the most efficient. In the context of the presented work, this investigation is part of future research.

The HMM training process extracts four probability values for each bass motion: 1) to begin the sequence, 2) to end the sequence, 3) to follow another bass motion (transition probability) and 4) to be present given a melody step (observation probability). The probabilities extracted by this process for each possible next bass motion is denoted with a vector of probabilities \vec{p}_m (one probability for each possible bass motion step) and will be utilised in the product of probabilities from all modules in Equation 1.

2.2 The voicing layout information module

In order to assign a bass voice to a chord, additional information is required that is relevant to the chords of the harmonisation. The voicing layout statistics that are considered for the modules of the presented methodology are the *inversions* and the *doublings* of chords. The inversions of a chord play an important role in determining how eligible is a chord’s pitch class to be a bass note, while the doublings indicate if additional ‘room’ between the

bass and the melody is required to fit doublings of specific pitch classes of the chords. For instance, the chord with pitch classes $[0, 4, 7]$ has three inversions, with each one having a bass note that corresponds to a different pitch class, e.g. $[60, 64, 67]$, $[64, 67, 72]$ or $[67, 72, 76]$, while, by considering the inversion prototype $[60, 64, 67]$ of the $[0, 4, 7]$ chord, there are four scenarios of single note doublings: $[60, 64, 67, 72]$, $[60, 64, 67, 76]$, $[60, 64, 67, 79]$ and $[60, 64, 67]$ (no-doubling scenario).

The voicing layout module of the harmonic learning system regarding chord inversions and note doublings, is trained through extracting relevant information from every (GCT) chord in pieces from a music idiom. Specifically, consider a GCT chord in the form $g = [r, \vec{t}]$, where r is the root of the chord in relation to the root of the key and \vec{t} is the vector describing the type of the chord. For instance, the I chord in any key is expressed as $g = [0, [0, 4, 7]]$ in the GCT representation, where 4 denotes the major third and 7 the perfect fifth. This GCT type is a set of integers, $\vec{t} = [t_1, t_2, \dots, t_n]$, where n is the number of type elements, that can be directly mapped to relative pitch classes (PCs). The statistics concerning chord inversion are expressed as the probability that each type element in g is the bass note of the chord, or

$$p_i = (v_1, v_2, \dots, v_n),$$

where $v_i, i \in \{1, 2, \dots, n\}$, is the probability that the element t_i is the bass note. Similarly, probabilities about note doublings are expressed through a probability vector

$$p_d = (d_1, d_2, \dots, d_n, s),$$

where $d_i, i \in \{1, 2, \dots, n\}$, is the probability that the pitch class t_i gets doubled, while there is an additional value, s , that describes the probability that there is no doubling of pitch classes. Table 2 exhibits the extracted statistics for inversions and note doublings for the most often met chords of the major Bach Chorales.

2.3 The melody-to-bass distance module

An important aspect of voice layout has to do with absolute range of chords in the chord sequences of an idiom, i.e. the absolute difference between the bass voice and the melody. Different idioms encompass different constraints and characteristics concerning this voicing layout aspect, according to several factors, e.g., the utilised instruments' range. The proposed methodology addresses this voicing layout aspect by capturing statistics about the *region* that the bass voice is allowed to move according to the melody. Therefore, histograms are extracted that describe the frequency of all melody-to-bass intervals found in a training dataset, as illustrated by the bars in the example in Figure 1.

However, interval-related information in the discussed context are used only as approximate indicators about the expected pitch height of the bass voice, while the exact intervals (bars in Figure 1) are referring to specific intervals and, additionally, they are scale-sensitive, e.g. differ-

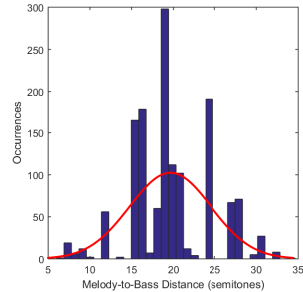


Figure 1. Histogram of pitch interval distances between melody and bass for a set of major Bach Chorales.

ent scales potentially produce different distributions of melody-to-bass intervals. Therefore, the “expected” bass pitch height is approximated by a normal distribution that is adjusted to fit the distribution of the melody-to-bass intervals observed in the dataset. Figure 1 illustrates the normal distribution that approximates the distributions of intervals for a collection of major Bach Chorales.

2.4 Combining all modules

The probabilities gathered from all the modules described hitherto are combined into a single value, computed as the product of all the probabilities from all the incorporated modules. To this end, for each GCT chord (C) in the composition every possible scenario of chord inversions, doublings and bass note pitch height, denoted by an index x , is generated. For each scenario (x), the product ($b_x(C)$) of all the modules discussed so far is computed, i.e. the bass motion ($p_{m_x}(C)$), the inversions ($p_{i_x}(C)$), doublings ($p_{d_x}(C)$) and melody-to-bass interval ($p_{h_x}(C)$):

$$b_x(C) = p_{m_x}(C) p_{i_x}(C) p_{d_x}(C) p_{h_x}(C). \quad (1)$$

Therefore, the best scenario (x_{best}) for the bass voice of chord C is found by: $x_{\text{best}} = \arg \max_x (b_x(C))$. The bass note motion probability is obtained by the HMM module analysed in Section 2.1 and it takes a value given by the vector \vec{p}_m according to the bass step it leads to.

3. EXPERIMENTAL RESULTS

The aim of the experimental process is to evaluate whether the proposed methodology efficiently captures the bass voice leading according to several factors related to the voice layout characteristics of each training idiom. Additionally, it is examined whether the separate trained modules, which constitute the overall system, statistically reveal aspects of each idiom that are more distinctive. A collection of eight datasets has been utilised for training and testing the capabilities of the proposed methodology, exhibited in Table 3.

These pieces are included in a music database with many diverse music idioms and it is developed for the purposes

GCT chord	relative PC	inversions	doublings
[0, [0, 4, 7]]	[0, 4, 7]	[0.74, 0.23, 0.02]	[0.68, 0.15, 0.08, 0.09]
[7, [0, 4, 7]]	[7, 11, 2]	[0.78, 0.22, 0.00]	[0.83, 0.02, 0.09, 0.06]
[5, [0, 4, 7]]	[5, 9, 0]	[0.65, 0.34, 0.01]	[0.46, 0.30, 0.11, 0.13]

Table 2. Probabilities for chord inversion (p_i) and note doublings (p_d) in the three most frequently used chords in the major Chorales of Bach.

Name (number)	Description
Bach Chorales (35)	a set of Bach chorales
Beatles (10)	set of songs from the band Beatles
Epirus (29)	traditional polyphonic songs from
Medieval (12)	fauxbourdon and organum pieces
Modal chorales (34)	15th-16th century modal chorales
Rembetika (22)	folk Greek songs
Stravinsky (10)	pieces composed by Igor Stravinsky
Tango (24)	pieces of folk tango songs

Table 3. Dataset description.

of the COINVENT project. For the presented experimental results, each idiom set includes from around 50 to 150 phrases. The Bach Chorales have been extensively utilised in automatic probabilistic melodic harmonisation [1, 7, 13, 16], while the polyphonic songs of Epirus [9, 11] and Rembetika [17] constitute datasets that have hardly been used in studies.

3.1 Cross-entropies for training and testing in all idiom combinations

The cross-entropy tests include the statistical modules that are independent of the GCT chords, i.e. HMM model and the melody-to-bass distance fitted distribution (will hereby be symbolised as mbd). Additionally, to examine the effect of the transition and the observation probabilities, the probabilities related to transitions of the bass (states transitions and will hereby be symbolised as tr) and the melody voice (observation transitions and will hereby be symbolised as mel) will be examined separately. The statistical combinations examined during the experimental evaluation process are: 1) the HMM model and the melody-to-bass distance fitted distribution probabilities (M^{all}), 2) only the bass voice transition probabilities from the HMM (M^{tr}), 3) only the melody observation probabilities from the HMM (M^{mel}) and 4) only the Melody-to-bass distance distributions (M^{mbd}).

Each idiom’s dataset is divided in two subsets, a training and a testing subset, with a proportion of 90% to 10% of the entire idiom’s pieces. The training subset of an idiom X is utilised to train the aforementioned modules, forming the trained model M_X , while the testing subset of the same idiom will be hereby denoted as D_X . For instance, the HMM trained with the Bach Chorales will be symbolised as M_{Bach} while its testing pieces will be symbolised as D_{Bach} . The evaluation of whether a model M_X predicts a subset D_X better than a subset D_Y is achieved through the cross-entropy measure. The measure of cross-entropy is utilised to provide an entropy value for a sequence from a dataset, $\{S_i, i \in \{1, 2, \dots, n\}\} \in D_X$, according to the

context of each sequence element, S_i , denoted as C_i , as evaluated by a model M_Y . The value of cross-entropy under this formalisation is given by

$$-\frac{1}{n} \sum_1^n \log P_{M_Y}(S_i, C_{i, M_Y}), \quad (2)$$

where $P_{M_Y}(S_i, C_{i, M_Y})$ is the probability value according to the examined scenarios of probabilities.

By comparing the cross-entropy values of a sequence X as predicted by two models, D_X and D_Y , we can assume which model predicts S better: the model that produces the *smaller* cross entropy value [8]. Smaller cross entropy values indicate that the elements of the sequence S “move on a path” with greater probability values. Tables 4 exhibits the cross-entropy values produced by the proposed model for the examined scenarios. The presented values are averages across 100 repetitions of the experimental process, with different random divisions in training and testing subsets (preserving a ratio of 90%-10% respectively for all repetitions). In every repetition the average cross entropy of all the testing sequences is calculated. The effectiveness of the combined proposed modules is indicated by the fact that most of the minimum values per row are on the main diagonal of the upper part of the matrix, i.e. where model M_X^{all} predicts D_X better than any other D_Y . A 10-fold cross-validation routine was also tested for splitting the dataset, however, replications of the experiment where different pieces in training and testing sets were used, gave considerably different results. The utilised experimental setup was providing similar results in several replications of the experiment.

It is evident that each module isolated does not produce lower values in the diagonal. Among the clearest isolated characteristics is the melody observations part of the HMM (M^{mel}), where 5 out of 8 diagonal elements are the lowest in their row. Thereby, these results indicate that the combination of all modules is a vital part for achieving better results.

3.2 Diversity in inversions and doublings of GCT chords

A straightforward comparison in statistics related to inversions and doublings between GCTs of different idioms is not possible for all idioms and all GCTs, since this information is harnessed on GCT sets that are in many cases different for different idioms. The differences in characteristics about voicing layout between different sets of GCTs that could be envisaged, relate to the *diversity* of the voicing layout scenarios that are used across different idioms.

	D_{Bach}	D_{Beattles}	D_{Epinus}	D_{Medieval}	D_{Modal}	$D_{\text{Rembetika}}$	$D_{\text{Stravinsky}}$	D_{Tango}
$M_{\text{Bach}}^{\text{all}}$	7.17	11.07	15.75	10.79	7.41	9.77	11.86	8.88
$M_{\text{Beattles}}^{\text{all}}$	9.75	7.82	15.97	14.86	9.77	8.27	7.64	9.01
$M_{\text{Epinus}}^{\text{all}}$	16.64	19.62	6.99	10.54	13.11	14.30	16.11	16.46
$M_{\text{Medieval}}^{\text{all}}$	10.96	17.56	7.68	7.47	8.49	12.46	16.18	12.63
$M_{\text{Modal}}^{\text{all}}$	9.27	15.94	15.04	10.96	8.39	10.89	15.32	10.72
$M_{\text{Rembetika}}^{\text{all}}$	8.73	8.56	13.65	11.79	8.22	7.11	7.80	8.29
$M_{\text{Stravinsky}}^{\text{all}}$	14.19	10.82	17.45	19.88	15.84	10.99	9.76	13.88
$M_{\text{Tango}}^{\text{all}}$	8.27	8.78	14.62	11.33	7.98	7.62	9.35	7.70
$M_{\text{Bach}}^{\text{H}}$	2.09	2.61	3.16	2.25	2.24	2.99	2.97	2.62
$M_{\text{Beattles}}^{\text{H}}$	3.51	2.33	2.47	3.30	2.88	1.82	2.28	2.20
$M_{\text{Epinus}}^{\text{H}}$	5.39	3.17	2.04	4.90	4.31	2.06	2.64	3.78
$M_{\text{Medieval}}^{\text{H}}$	2.73	2.92	1.97	2.33	2.33	2.49	2.74	3.11
$M_{\text{Modal}}^{\text{H}}$	2.87	2.92	2.82	2.41	3.32	2.79	2.73	3.07
$M_{\text{Rembetika}}^{\text{H}}$	4.11	2.66	1.90	3.53	3.21	1.67	1.88	2.62
$M_{\text{Stravinsky}}^{\text{H}}$	5.44	3.98	2.51	4.51	4.73	2.63	3.50	4.50
$M_{\text{Tango}}^{\text{H}}$	3.11	2.16	2.82	2.98	3.02	1.88	2.55	2.12
$M_{\text{Bach}}^{\text{mel}}$	1.79	2.14	2.28	1.95	1.85	2.34	2.44	2.15
$M_{\text{Beattles}}^{\text{mel}}$	2.34	1.92	2.09	2.26	1.93	1.65	1.87	1.86
$M_{\text{Epinus}}^{\text{mel}}$	2.72	2.43	1.42	2.21	2.43	1.72	1.74	2.59
$M_{\text{Medieval}}^{\text{mel}}$	2.54	3.32	2.15	2.13	2.50	2.36	2.51	3.04
$M_{\text{Modal}}^{\text{mel}}$	2.68	2.60	2.57	2.64	2.36	2.12	2.55	2.59
$M_{\text{Rembetika}}^{\text{mel}}$	2.81	2.13	1.86	2.39	2.20	1.37	2.17	2.00
$M_{\text{Stravinsky}}^{\text{mel}}$	3.77	3.12	2.29	3.85	3.39	2.83	2.53	3.77
$M_{\text{Tango}}^{\text{mel}}$	2.33	1.86	1.94	2.36	1.90	1.48	2.17	1.72
$M_{\text{Bach}}^{\text{mbd}}$	3.58	6.51	10.50	6.77	3.55	4.45	5.65	4.25
$M_{\text{Beattles}}^{\text{mbd}}$	4.90	4.24	12.17	10.13	5.63	4.72	3.90	5.38
$M_{\text{Epinus}}^{\text{mbd}}$	9.03	14.89	3.51	4.14	6.83	10.31	12.04	10.34
$M_{\text{Medieval}}^{\text{mbd}}$	6.10	13.05	3.77	3.93	4.57	7.72	10.82	7.15
$M_{\text{Modal}}^{\text{mbd}}$	4.44	11.53	10.35	6.48	3.47	6.18	9.70	5.63
$M_{\text{Rembetika}}^{\text{mbd}}$	3.79	4.80	10.59	6.80	3.92	4.11	4.32	4.20
$M_{\text{Stravinsky}}^{\text{mbd}}$	5.87	4.56	12.91	12.08	8.00	6.18	4.67	6.73
$M_{\text{Tango}}^{\text{mbd}}$	3.64	5.35	10.38	6.56	3.70	4.12	4.78	4.19

Table 4. Mean values of cross-entropies for all pairs of datasets, for all the combination of all probabilities, as well as in isolation concerning previous bass motion, melody motion and bass-to-melody distance.

Along these lines, the question would be: are there more diverse chord expressions regarding inversions and doublings – regardless of which chords (GCTs) – in the chorales of Bach, than in the modal chorales? The *diversity* in a discrete probability distribution (like the ones displayed in the examples of Table 2) is measured by the Shannon information entropy [21] (SIE). The SIE reflects the diversity in possibilities described by discrete probability distribution, with higher SIE values indicating a more random distribution with more diverse / less expectable outcomes. Therefore, by measuring the SIE values of all GCTs and comparing them for every pair of idioms, it can be concluded whether some idioms have richer possibilities for the voicing layouts of chords than others.

Table 5 exhibits the results of a test in the statistical significance in differences between the SIE values in every pair of idioms. The upper-diagonal elements concern inversions, while lower-diagonal elements doublings. A value of +1 indicates that the GCTs in the idiom of the row are statistically significantly more diverse in their voicing layout – according to the mean SIE values – than the ones in the idiom of the column. A -1 value indicates the opposite, while a 0 value indicates no statistically significant

difference. The statistical significance is measured through a two-sided Wilcoxon [23] rank sum test, which is applied on the SIE values of all GCT voicing layout distributions for every idiom. The statistical significance test in statistics related to voice layout reveal that few datasets are significantly superior or inferior regarding their diversity.

3.3 Example compositions

The proposed bass voice leading methodology was utilised in an “off-line” mode to produce two examples. The term “off-line” indicates the fact that the system was used to generate a single description for the bass voice leading on a given set of chords (in GCT representation [2] produced by a probabilistic chord-generation model [10]). This means that if no inversion of the predetermined chord can satisfy the requirements of the bass voice leading, then the system simply selected the most probable inversion of this chord, regardless of the bass voice leading indication. The bass voice for the generated examples was selected using the *argmax* function mentioned in Section 2.4, which allows the reflection of some typical idiom characteristics, even though such an approach does not necessarily guaranty interestingness [3] (since the most “expected” scenario is fol-

	S_{Bach}	$S_{Beattles}$	S_{Epirus}	$S_{Medieval}$	S_{Modal}	$S_{Rembetika}$	$S_{Stravinsky}$	S_{Tango}
S_{Bach}	0	1	1	-1	1	1	1	1
$S_{Beattles}$	0	0	0	-1	0	0	0	0
S_{Epirus}	0	0	0	-1	0	0	0	0
$S_{Medieval}$	-1	-1	-1	0	1	1	1	1
S_{Modal}	0	0	0	1	0	1	1	0
$S_{Rembetika}$	0	-1	0	1	0	0	0	0
$S_{Stravinsky}$	0	0	1	0	1	1	0	0
S_{Tango}	0	0	0	1	0	0	0	0

Table 5. Statistical significance of differences in the diversity of inversions (upper diagonal) and doublings (lower diagonal). Statistically significant superiority of diversity in the row dataset is exhibited with a +1, of the column dataset with -1, while 0 indicates no statistical significance in diversity differences.

lowed). The intermediate voices were manually adjusted by a music expert.

The presented examples (Figure 2) include two alternative harmonisations of a Bach Chorale melody with both the chord generation and the bass voice leading systems trained on sets of (a) the Bach Chorales and (b) polyphonic songs from Epirus. In the case of the Bach chorale, the system made erroneous bass voice assignments in the second bar that create consecutive anti-parallel octaves between the outer voices (due to the chord incompatibility problem discussed above)¹. The harmonisation in the style of the polyphonic songs from Epirus indeed preserves an important aspect of these pieces: the drone note.

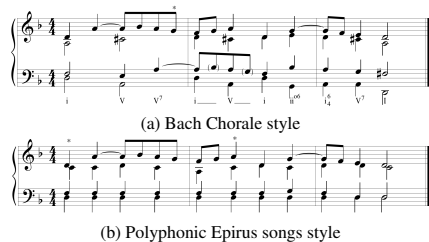


Figure 2. Harmonisation examples in two different styles. Chord sequences in the GCT representation were previously produced by another probabilistic system.

4. CONCLUSIONS

This paper presented a modular methodology for determining the bass voice leading in automated melodic harmonisation given a melody voice and a sequence of chords. In this work it is assumed that harmony is not solely the expression of a chord sequence, but also of harmonic movement for all voices that comprise the harmonisation. The presented work focuses on generating the bass voice on a given sequence of chords by utilising information from the

¹ Another voice-leading issue occurs at the first beat of the 3rd bar, where the D in the 2nd voice is introduced as unprepared accented dissonance. Note that the parenthesised pitches in the 3rd voice (bar 2) were introduced manually (not by the system) to create imitation.

soprano /melody voice and other statistics that are related to the layout of the chords, captured by different statistical modules. Specifically, a hidden Markov model (HMM) is utilised to determine the most probable movement for the bass voice (hidden states), by observing the soprano movement (set of observations), while additional voicing layout characteristics of the incorporated chords are considered that include distributions on the distance between the bass and the melody voice and statistics regarding the inversions and doublings of the chords in the given chord sequence.

Experimental results evaluate that the learned statistical values from an idiom's data are in most cases efficient for capturing the idiom's characteristics in comparison to others. Additionally, similar tests were performed for each statistical module of the model in isolation, a process that revealed whether some characteristics of the examined idioms are more prominent than others. Furthermore, preliminary music examples indicate that the proposed methodology indeed captures some of the most prominent characteristics of the idioms it is being trained with, despite the fact that further adjustments are required for its application in melodic harmonisation.

5. ACKNOWLEDGEMENTS

This work is funded by the COINVENT project. The project COINVENT acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 611553. The authors would like to thank Costas Tsougras for his assistance in preparing the presented musical examples.

6. REFERENCES

- [1] Moray Allan and Christopher K. I. Williams. Harmonising chorales by probabilistic inference. In *Advances in Neural Information Processing Systems 17*, pages 25–32. MIT Press, 2004.
- [2] Emiliós Cambouropoulos, Maximos Kaliakatsos-Papakostas, and Costas Tsougras. An idiom-independent representation of chords for compu-

- tational music analysis and generation. In *Proceeding of the joint 11th Sound and Music Computing Conference (SMC) and 40th International Computer Music Conference (ICMC)*, ICMC-SMC 2014, 2014.
- [3] Tom Collins. *Improved methods for pattern discovery in music, with applications in automated stylistic composition*. PhD thesis, The Open University, 2011.
- [4] Patrick Donnelly and John Sheppard. Evolving four-part harmony using genetic algorithms. In *Proceedings of the 2011 International Conference on Applications of Evolutionary Computation - Volume Part II, EvoApplications'11*, pages 273–282, Berlin, Heidelberg, 2011. Springer-Verlag.
- [5] Kemal Ebcioglu. An expert system for harmonizing four-part chorales. *Computer Music Journal*, 12(3):43–51, 1988.
- [6] David Huron. Voice denumerability in polyphonic music of homogeneous timbres. *Music Perception*, pages 361–382, 1989.
- [7] Michael I. Jordan, Zoubin Ghahramani, and Lawrence K. Saul. Hidden markov decision trees. In Michael Mozer, Michael I. Jordan, and Thomas Petsche, editors, *NIPS*, pages 501–507. MIT Press, 1996.
- [8] Dan Jurafsky and James H. Martin. *Speech and language processing*. Prentice Hall, New Jersey, USA, 2000.
- [9] M. Kaliakatsos-Papakostas, A. Katsiavalos, C. Tsougras, and E. Cambourooulos. Harmony in the polyphonic songs of epirus: Representation, statistical analysis and generation. In *4th International Workshop on Folk Music Analysis (FMA) 2014*, June 2011.
- [10] Maximos Kaliakatsos-Papakostas and Emiliós Cambourooulos. Probabilistic harmonisation with fixed intermediate chord constraints. In *Proceeding of the joint 11th Sound and Music Computing Conference (SMC) and 40th International Computer Music Conference (ICMC)*, ICMC-SMC 2014, 2014.
- [11] Kostas Liolis. *To Epirótiko Polyphónikó Tragoúdi (Epirus Polyphonic Song)*. Ioannina, 2006.
- [12] Dimos Makris, Maximos Kaliakatsos-Papakostas, and Emiliós Cambourooulos. A probabilistic approach to determining bass voice leading in melodic harmonisation. In Tom Collins, David Meredith, and Anja Volk, editors, *Mathematics and Computation in Music*, volume 9110 of *Lecture Notes in Computer Science*, pages 128–134. Springer International Publishing, 2015.
- [13] Leonard C. Manzara, Ian H. Witten, and Mark James. On the entropy of music: An experiment with bach chorale melodies. *Leonardo Music Journal*, 2(1):81–88, January 1992.
- [14] Francois Pachet and Pierre Roy. Formulating constraint satisfaction problems on part-whole relations: The case of automatic musical harmonization. In *Proceedings of the 13th European Conference on Artificial Intelligence (ECAI 98)*, pages 1–11. Wiley-Blackwell, 1998.
- [15] Francois Pachet and Pierre Roy. Musical harmonization with constraints: A survey. *Constraints*, 6(1):7–19, January 2001.
- [16] Jean-François Paiement, Douglas Eck, and Samy Bengio. Probabilistic melodic harmonization. In *Proceedings of the 19th International Conference on Advances in Artificial Intelligence: Canadian Society for Computational Studies of Intelligence, AI'06*, pages 218–229, Berlin, Heidelberg, 2006. Springer-Verlag.
- [17] Risto Pekka Pennanen. The development of chordal harmony in greek rebetika and laika music, 1930s to 1960s. *British Journal of Ethnomusicology*, 6(1):65–116, 1997.
- [18] Somnuk Phon-amnuaisuk and Geraint A. Wiggins. The four-part harmonisation problem: A comparison between genetic algorithms and a rule-based system. In *In proceedings of the AISB99 symposium on musical creativity*, pages 28–34. AISB, 1999.
- [19] Martin Rohmeier and Thore Graepel. Comparing feature-based models of harmony. In *Proceedings of the 9th International Symposium on Computer Music Modelling and Retrieval*, pages 357–370, 2012.
- [20] M. Schorlemmer, A. Smaill, K.U. Kühnberger, O. Kutz, S. Colton, E. Cambourooulos, and A. Pease. Coinvent: Towards a computational concept invention theory. In *5th International Conference on Computational Creativity (ICCC) 2014*, June 2014.
- [21] C. E Shannon. A mathematical theory of communication. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5:3–55, January 2001.
- [22] Raymond P. Whorley, Geraint A. Wiggins, Christophe Rhodes, and Marcus T. Pearce. Multiple viewpoint systems: Time complexity and the construction of domains for complex musical viewpoints in the harmonization problem. *Journal of New Music Research*, 42(3):237–266, September 2013.
- [23] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83, 1945.
- [24] Liangrong Yi and Judy Goldsmith. Automatic generation of four-part harmony. In Kathryn B. Laskey, Suzanne M. Mahoney, and Judy Goldsmith, editors, *BMA*, volume 268 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2007.

AN ITERATIVE MULTI RANGE NON-NEGATIVE MATRIX FACTORIZATION ALGORITHM FOR POLYPHONIC MUSIC TRANSCRIPTION

Anis Khlif

École des Mines ParisTech, France
anis.khlif@mines-paristech.fr

Vidhyasaharan Sethu

University of New South Wales, Australia
v.sethu@unsw.edu.au

ABSTRACT

This article presents a novel iterative algorithm based on Non-negative Matrix Factorisation (NMF) that is particularly well suited to the task of automatic music transcription (AMT). Compared with previous NMF based techniques, this one does not aim at factorizing the time-frequency representation of the entire musical signal into a combination of the possible set of notes. Instead, the proposed algorithm proceeds iteratively by initially decomposing a part of the time-frequency representation into a combination of a small subset of all possible notes then re-investing this information in the following step involving a large subset of notes. Specifically, starting with the lowest octave of notes that is of interest, each iteration increases the set of notes under consideration by an octave. The resolution of a lower dimensionality problem used to properly initialize matrices for a more complex problem, results in a gain of some percent in the transcription accuracy.

1. INTRODUCTION

The term Automatic Music Transcription (AMT) refers to the task of designing a system that automatically transposes an acoustic signal into a written format that can be read by a musician e.g. sheet music. In Western music, the basic unit of this transposition is the note, which is partly defined by its duration and its pitch. When more than one note can occur at the same time, the music is said to be polyphonic. Further, each instrument has its own harmonic pattern that is time-dependent for each of its notes. Indeed, the spectral content during the onset part of a note is different from the one during the sustain or fading parts. AMT of polyphonic musics amounts to tracking the fundamental frequencies among a mixture of musical events with possibly overlapping harmonics. Many approaches have been proposed but the results are still unsatisfactory compared to what can be achieved by a human expert [5]. Lately, techniques like NMF [17] [16] [7] and Probabilistic Latent

Component Analysis (PLCA) [4] [18] have gained great interest since they have proved very efficient in bringing forward the underlying structure of musical data. Both are conceptually linked and have been shown equivalent under certain formulations [8]. They provide a framework under which the transcription can be formulated as a cost-function minimization problem, which are deeply studied problems and many algorithms exist to solve them. However, these algorithms (such as gradient descent, expectation maximization, alternating least-squares, etc...) suffer from major flaws. They offer no guarantees of finding a global minimum (if any) in general, and can easily get stuck in local ones. On top of this, they are highly sensitive to initial conditions and an improper initialization can lead to bad results [6] [1]. These issues are great liabilities for AMT because the intricate nature of harmonically related sounds results in the existence of many local minima which in turn increases the chance of an incorrect transcription.

In this paper we present an NMF-based algorithm tailored for the task of AMT, showing increased robustness with respect to the issues of finding proper initialization parameters and avoiding irrelevant local minima.

2. THE NMF FRAMEWORK

2.1 General overview

The different steps of the algorithm are presented in Figure 1. First, the time-frequency representation of the signal (spectrogram) is computed by applying a Constant Q Transform (CQT) on successive time windows. Then the proposed IMRNMF algorithm is applied to the spectrogram to produce a matrix representing the activation of each note across time. This matrix is then post-processed to extract chunks representing potential notes which are then weighted before being truly acknowledged as a note and transcribed.

NMF aims at representing a non-negative signal as an additive synthesis of events taken from a finite dictionary. The original signal is then represented by the activation at each time of a subset of these events. If the signal lends itself to such description, the decomposition will likely be meaningful in the sense that it will bring out some of the underlying structure. In the case of AMT, the decomposition of the music into events that can be assimilated to notes, would be most desirable. A time-frequency repre-



© Anis Khlif, Vidhyasaharan Sethu.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Anis Khlif, Vidhyasaharan Sethu. "An Iterative Multi Range Non-Negative Matrix Factorization algorithm for polyphonic music transcription", 16th International Society for Music Information Retrieval Conference, 2015.

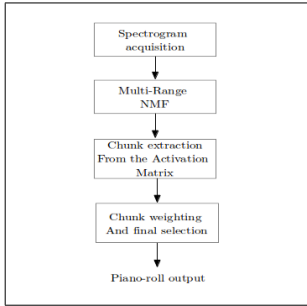


Figure 1: Overall algorithm.

sensation, like the spectrogram (which is a matrix containing the amplitude spectrum for a sequence of time windows) is an example of additive data where the sources would be constituted by the amplitude spectra of the different notes composing it. As mentioned, we will consider that the time-frequency representation is obtained via a CQT, which allows all frequencies of interest for all notes to be contained in the same number of frequency bins (in the time frequency representation) regardless of the octave or the note unlike the standard Discrete Fourier Transform.

More formally, given the spectrogram $Y \in \mathbb{R}_+^{N \times T}$, where N is the number of frequency bins and T the number of temporal frames, and given $K \leq N, T$; find $W \in \mathbb{R}_+^{N \times K}$ the spectral dictionary matrix, and $H \in \mathbb{R}_+^{K \times T}$ such that

$$Y \approx WH \quad (1)$$

Where, Y denotes the spectrogram obtained from the CQT, and which is decomposed into weighted sums of a finite set of notes whose spectra constitute the columns of W .

This decomposition does not have an exact solution and consequently the typical approach is to find a solution that minimises a suitable cost function, $C_Y(W, H)$ with the constraints that the elements of W and H are positive. Historically, as introduced by Paatero [15] the canonical norm of the matrices difference: $\|Y - WH\|$ was taken as a cost function. Incidentally, a factorization is inherently dependent on the cost function used to weight the reconstitution. As a result, the choice of a relevant cost function to increase the accuracy of the decomposition has been largely studied and yielded significant increases in the results. In the next section we review some of the key principles driving current efforts to enhance the transcription through NMF related techniques.

2.2 Achieving a good factorization

The best factorization we could hope for, would express Y as the activation of spectral templates that correspond exactly to the ones of the notes present in the excerpt. That implies especially, that no existing note be expressed as the

sum of two or more elements (columns) of the dictionary W , (no false detection), or that no combination of two or more notes be expressed by a single element (no deletion). Such issues are referred to as cross-row talk. A common response to cross-row talk is to try to increase the sparsity of the decomposition matrices, and especially the columns of H . (A vector is said to be sparse when most of its elements are zeros). The energy is concentrated in a few units which are used to represent typical data vectors. Having a control over sparsity provides more robustness in "real-life" situations where the number of sources is not known by advance and a higher rank than needed is fixed for the decomposition matrix.

Controlling the sparsity is mainly achieved by choosing a suitable cost function C_Y and estimation methods that allow desirable properties to be enforced on W and H . Although, the task of finding a minimum for C_Y is not easy since the problem is often ill-posed, the reformulation of the factorization problem in terms of approaches such as Convex Quadratic Programming [7] [19] [11] provides elegant frameworks to naturally introduce new cost functions (with regularization parameters), or enforce relevant constraints on W and H .

The control over sparsity can be explicit. In [12], Hoyer develops algorithm to enforce constant predefined sparsities s_w, s_h over W and H . Such conditions are not realistic in real-life situations for audio data since the degree of polyphony can evolve throughout the excerpt. In [11], Heiler and Schnörr, give a formulation of the factorization as a second order cone programming problem, enabling them to enforce only boundary conditions on the sparsities. In [1], an adaptation of the ALS algorithm called Alternating Hoyer-Constrained Least Squares is proposed. However, this way of enforcing sparsity is often too restrictive in the case of musical data where the degree of polyphony is free to evolve during time, on top of the fact that we do not have prior knowledge on it. Consequently, we would prefer a softer, implicit control over sparsity. In such cases, it is often achieved through cost functions that are expressed in a form where the variation of a parameter provides an input to indirectly affect sparsity. In [7] the cost function is defined by

$$C_y = \frac{1}{2} \|Y - WH\|_2^2 + \lambda_1 \|H\|_1 + \frac{\lambda_2}{2} \|H\|_2^2 \quad (2)$$

The coefficient λ_1 weights the importance given to sparse vectors against a good reconstitution, and λ_2 is a Tikhonov regularization parameter. Other successful approaches have considered a class of divergences called β -divergences as cost functions [10], which were successfully applied to AMT in [7]. $d_\beta(Y|W, H)$ is defined by:

$$d_\beta(Y|W, H) = \begin{cases} Y \otimes \log \frac{Y}{WH} - Y + WH & \beta = 1 \\ \frac{Y}{WH} - \log \frac{Y}{WH} - 1 & \beta = 0 \\ \frac{1}{\beta(\beta-1)} (Y^\beta + (\beta-1)(WH)^\beta - \beta Y \otimes (WH)^{\beta-1}) & \text{else} \end{cases} \quad (3)$$

Where the divisions, the logarithm and the powers have to be understood element-wise, \otimes is the element-wise product, and $\mathbf{1}$ the matrix containing only ones. The choice

of β provides an indirect control over sparsity. It can be noted that in the case of $\beta = 2$ it reduces to the Euclidean distance, and in the case $\beta = 1$ to the KL-div KL divergence, which has been found to promote sparsity [17]. The minimization of both those cost functions can be achieved through multiplicative update rules given in [10] and [7]. This is the cost function which has been adopted in the proposed method.

Finally, all the algorithms mentioned are highly sensitive to initial conditions and perform poorly when dimension and the density of local minima increase. In the case of AMT, initializing the spectral dictionary matrix W so that the elements (columns) are structurally relevant, improves the factorization a great deal. In [16], the columns of W are initialized with one for each note at harmonic positions and zeros elsewhere. It makes W relevant for the transcription and straightforward to associate to a note. While it is not too difficult to see how W can be initialized, it is much less obvious for H .

In the next section, we present a versatile algorithm to perform the factorization which can be used with any update rules, enhances the sparsity and gives element of answer as to how initialize H leading to increased robustness.

3. THE PROPOSED FACTORIZATION ALGORITHM

3.1 Principle

The proposed algorithm performs an iterative factorization of the spectrogram by initially starting with a single octave of notes prior to incrementing it by an octave in each subsequent iteration. The algorithms performs by starting from the lowest octave, and by including one higher octave at each step until the whole range of note is covered. Let $S = \{n_0, \dots, n_{K-1}\} \in \mathbb{N}^K$ be an interval of integers containing the midi notes considered. The i -th range is the subset of S defined by $r_i = \{n_0, \dots, n_{12i-1}\}$.

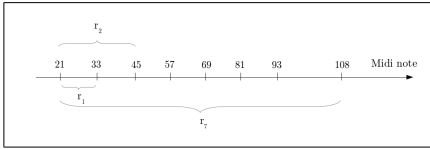


Figure 2: Cutting of the midi scale in ranges.

Only considering the notes lying in this range comes down to focusing on subregions, in terms of frequency and notes, of the decomposition matrices defined as follows.

$$Y^{(i)} \approx W^{(i)} H^{(i)} \quad (4)$$

with:

$$Y^{(i)} = Y_{[l_b, u_b^i], \bullet} \quad (5)$$

$$W^{(i)} = W_{[l_b, u_b^i], [l_s, u_s^i]} \quad (6)$$

The columns of W and the rows of H , indexed by the sources, are restricted to the subset $\{l_s, \dots, u_s^i\}$ where l_s denotes the source of the lower note and u_s^i the source of the higher note of the i -th range. We have the following equalities: $l_s = n_0$ and $u_s^i = n_{12i-1}$. The rows of W and Y representing the frequency bins are restricted to the subregion $\{l_b, \dots, u_b^i\}$ where l_b designs the lower frequency bin associated with the fundamental frequency of the lower note in the range, and u_b^i the upper bound for the frequency bins associated with the fundamental frequency of the higher note in the i -th range. As previously mentioned, the spectrogram is computed with a CQT, therefore we can note that the semitone resolution, b, i.e., the number of bins associated with a single semitone is a constant. The superscript (i) denotes the restriction of a matrix to the i -th range. With this notation, we can express the boundaries as: $l_b = b(n_0 - 1) + 1$ and $u_b^i = b(n_{12i-1})$. All the temporal frames are considered at each step of the factorization, this is noted \bullet .

As it has been said, any multiplicative update rule can be used with this approach. Specifically, in the work reported in this paper, the update rules (8) and (9) for the KL divergence are applied as follows to $H^{(i)}$ and the submatrix $W^{(i)}$.

$$H^{(i)} \leftarrow H^{(i)} \otimes \frac{tW^{(i)}(Y^{(i)} \otimes (W^{(i)}H^{(i)})^{\beta-2})}{tW^{(i)}(W^{(i)}H^{(i)})^{\beta-1}} \quad (7)$$

$$W^{(i)} \leftarrow W^{(i)} \otimes \frac{(Y^{(i)} \otimes (W^{(i)}H^{(i)})^{\beta-2})^t H^{(i)}}{(W^{(i)}H^{(i)})^{\beta-1} H^{(i)}} \quad (8)$$

Then $H^{(i+1)}$ is initialized as follows (see 3):

$$\begin{cases} H_{[l_s, u_s^i], \bullet}^{(i+1)} = H^{(i)} \\ H_{[u_s^i, u_s^{i+1}], \bullet}^{(i+1)} = \text{random positive matrix} \end{cases} \quad (9)$$

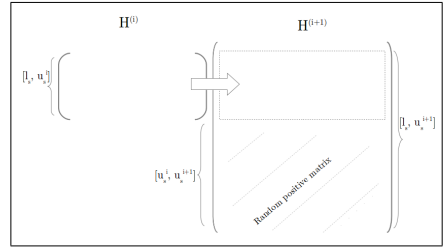


Figure 3: Initialization of $H^{(i+1)}$ from $H^{(i)}$.

Figures depicting the evolution of the activation matrix throughout the different steps are shown in section 5.

3.2 Motivation and advantages

This method has been designed as a way to compensate for some of the weaknesses of NMF applied to AMT, principally being having to use more potential sources than

strictly necessary in the decomposition (which can cause confusion in the factorization, hence the necessity of enforcing sparsity), and the high spectral similarity between certain combination of harmonically related notes, which added to the high number of sources is likely to increase the probability of falling into a local minimum. Starting from the lowest octave, helps secure a sound bass-line and avoid confusing notes with weak fundamental with their upper octave counterpart (octave problem); as it is likely to happen in the usual implementation since low notes often have a weak fundamental. Incrementing the set of notes by a single octave is also a step in this direction, in order to limit as much as possible the risks of mistaking a note for one of its harmonically related counterparts. Beside, limiting the number of sources reduces the dimension of the problem and heuristically, the risks of falling into a local minimum. Re-investing knowledge in the next steps of the factorization helps converge toward a better minimum by ensuring convergence on growing subspaces, where confusion is less likely. The resulting activation matrix is much sparser, and much easier to post-process because of the more distinct activation peaks.

An additional advantage of the proposed method is that it allows for different treatments on the parts of the spectrogram that are factorized. For instance, it allows for the definition of octave-based tolerance thresholds in terms of amplitude or spatial repartition (peaks with a maximum value under a threshold or ranging on less than a given number of frames will be discarded). Various works in the fields of psychoacoustics and acoustic signal processing showed that such treatment is of the utmost importance in order to reliably weight and perform competitive selection between acoustic events distributed across a large frequency span and with different amplitudes [13] [20] [14].

4. BACK-END TRANSCRIPTION

The back-end transcription limits itself to the mere detection of activation events in H , since the initialization of W made straightforward the association between events and notes. H having previously been normalized we applied a threshold-based onset detection, allowing to debit activation matrix rows into chunks that can further along be weighted and sieved before being labelled as note. Those chunks are bits of the activation matrix defined by: the midi note (the row number), the onset time and the offset time. The computation of the onsets is performed by applying an adaptive thresholding on the first order differential vector of each row of H as suggested in [2]. The thresholding value is based on the mean of the half-wave rectified first order differential signal on the 100 neighbouring frames. The onset is defined as the first frame for which the amplitude is superior than 0.2 times the thresholding value (it has experimentally been found as a good value).

A score on the chunks was defined in order to perform a post-selection of the chunk and screen out the ones that are very likely false positives. This cost function is based on features of the chunks considered as indicators of the

probability of this chunk to represent a true positive. This features are: the length of the chunk l , the maximum value of the amplitude within this chunk m , the value of the first order differential of the signal at the onset time (representing the steepness of the onset) d , and the energy of the signal e within the chunk against the cumulated energy of the signal of lower harmonics during the same time range e_l . The score of a chunk is defined as:

$$S = (1 - \exp(\frac{-l}{c_1}))(1 - \exp(\frac{-m}{c_2})) \\ (1 - \exp(\frac{-d}{c_3}))(1 - \exp(\frac{-e}{c_1 e_l})) \quad (10)$$

where c_1 , c_2 , c_3 , and c_4 are arbitrary constants. For the tests we used $(c_1, c_2, c_3, c_4) = (8, 0.1, 0.03, 0.66)$ and only chunks with a score higher than 0.2 were kept. These values were experimentally determined as reasonable and were kept fixed for the totality of our test. No music-specific fine-tuning was performed.

5. EXPERIMENTAL RESULTS

Tests were performed on the MAPS ENSTDkCl database [9] which is composed exclusively of piano recordings with a wide variety of polyphony, genre, tempo, and rhythm. The set of notes taken into account ranges between the midi notes 21 and 108. The spectrogram is computed by a CQT algorithm with sixty bins per octave to be robust to frequency shifts around the theoretical peak position. beta divergence cost function, with $\beta = 1$ (KL-divergence) was chosen for all matrix factorisations. The matrix W is kept fixed during the step-by-step factorization then, an additional standard NMF is performed with initialization from previous results. Our Iterative Multi Range Non-negative Matrix Factorization (IMRNMF) system is compared against an NMF-based system without the range-by-range factorization but the same back-end transcription algorithm; and the winning algorithm of the MIREX 2013 competition in Multi-F0 note tracking and Multi-F0 note estimation based on Shift Invariant Probabilistic Latent Component Analysis (*SI_PLCA*) [3]. The matrix W is initialized offline using the array provided with the *SI_PLCA* source code which consists of pre-extracted and pre-shifted spectral templates for various instruments. An onset-based metric is used with a 50 ms tolerance.

The transcription is performed on the first 30 seconds of each track in the database. The thresholding and weighing constants used in the back-end transcription as well as in the IMRNMF are kept fixed during the whole test independently of the extract being processed, and even better results can be achieved with a case-by-case fine tuning of these constants, based on parameters such as genre and tempo. Below are shown illustrative examples of the evolution of the activation matrix on the MAPS _MUS_schu_143_3_ENSTDkCl track.

Method	Accuracy	F-measure
<i>NMF</i>	0.38	0.55
<i>SI_PLCA</i>	0.37	0.53
IMRNMF	0.52	0.69

Table 1: Comparative results on the MAPS ENSTDkCl database.

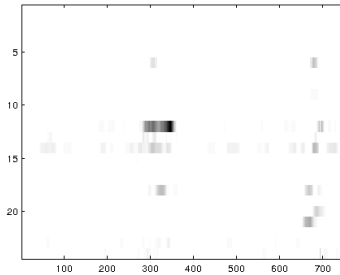


Figure 4: $H^{(2)}$ after the first 2 steps

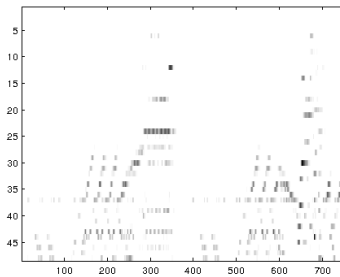


Figure 5: $H^{(4)}$ after the first 4 steps

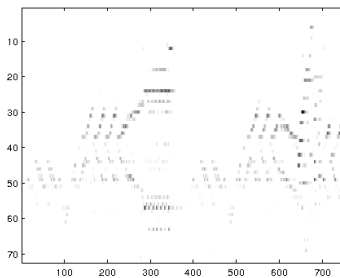


Figure 6: $H^{(6)}$ after the first 6 steps

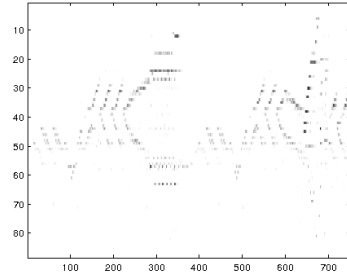


Figure 7: Final output of the factorization

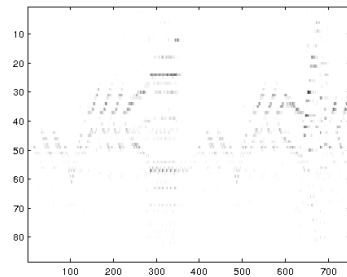


Figure 8: H obtained with *SI_PLCA*

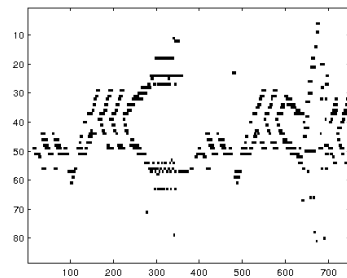


Figure 9: Backend transcription output

6. CONCLUSION AND FUTURE WORK

A novel Iterative Multi-Range Non-negative Matrix Factorisation (IMRNMF) based algorithm for automatic music transcription is presented in this paper. At the cost of increased computational requirements, though still perfectly accessible, the proposed system leads to an increase in transcription accuracy compared to the top-performing existing algorithms. This increase may be

better explained by the increased sparsity of the activation matrix. The improved sparsity is most likely due to the proposed algorithm finding better local minima to the cost function when compared to the traditional NMF. While a number of parameters in the proposed systems are empirically determined at this stage (thresholding constants, weighting parameters, chunk-wise cost function in the final decision process...), a more data-driven approach to estimating them may lead to even better performance and will be addressed in future work.

Acknowledgements. Many thanks to Emmanouil Benetos for providing the sources of the Shift-Invariant Probabilistic Latent Component Analysis algorithm.

7. REFERENCES

- [1] Russell Albright, James Cox, David Duling, Amy N Langville, and C Meyer. Algorithms, initializations, and convergence for the nonnegative matrix factorization. Technical report, Tech. rep. 919. NCSU Technical Report Math 81706. <http://meyer.math.ncsu.edu/Meyer/Abstracts/Publications.html>, 2006. url:<http://citeseerx.ist.psu.edu/viewdoc/download>, 2006.
- [2] Juan Pablo Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B Sandler. A tutorial on onset detection in music signals. *Speech and Audio Processing, IEEE Transactions on*, 13(5):1035–1047, 2005.
- [3] Emmanouil Benetos, Srikanth Cherla, and Tillman Weyde. An efficient shift-invariant model for polyphonic music transcription. In *6th International Workshop on Machine Learning and Music*, 2013.
- [4] Emmanouil Benetos and Simon Dixon. A shift-invariant latent variable model for automatic music transcription. *Computer Music Journal*, 36(4):81–94, 2012.
- [5] Emmanouil Benetos, Simon Dixon, Dimitrios Giannoulis, Holger Kirchhoff, and Anssi Klapuri. Automatic music transcription: Breaking the glass ceiling. In *ISMIR*, pages 379–384. Citeseer, 2012.
- [6] Michael W Berry, Murray Browne, Amy N Langville, V Paul Pauca, and Robert J Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, 52(1):155–173, 2007.
- [7] Arnaud Dessein, Arshia Cont, Guillaume Lemaître, et al. Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence. In *ISMIR-11th International Society for Music Information Retrieval Conference*, pages 489–494, 2010.
- [8] Chris Ding, Tao Li, and Wei Peng. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. *Computational Statistics & Data Analysis*, 52(8):3913–3927, 2008.
- [9] Valentin Emiya, Roland Badeau, and Bertrand David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(6):1643–1654, 2010.
- [10] Cédric Févotte and Jérôme Idier. Algorithms for non-negative matrix factorization with the β -divergence. *Neural Computation*, 23(9):2421–2456, 2011.
- [11] Matthias Heiler and Christoph Schnörr. Learning sparse representations by non-negative matrix factorization and sequential cone programming. *The Journal of Machine Learning Research*, 7:1385–1407, 2006.
- [12] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *The Journal of Machine Learning Research*, 5:1457–1469, 2004.
- [13] Anssi Klapuri. Sound onset detection by applying psychoacoustic knowledge. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 6, pages 3089–3092. IEEE, 1999.
- [14] Brian CJ Moore. *An introduction to the psychology of hearing*. Brill, 2012.
- [15] Pentti Paatero and Unto Tapper. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126, 1994.
- [16] Stanisław A. Raczynski, Nobutaka Ono, and Shigeki Sagayama. Multipitch analysis with harmonic nonnegative matrix approximation. In *ISMIR 2007, 8th International Conference on Music Information Retrieval*, pages 381–386, 2007.
- [17] Paris Smaragdis and Judith C Brown. Non-negative matrix factorization for polyphonic music transcription. In *Applications of Signal Processing to Audio and Acoustics, 2003 IEEE Workshop on.*, pages 177–180. IEEE, 2003.
- [18] Emmanuel Vincent, Nancy Bertin, and Roland Badeau. Harmonic and inharmonic nonnegative matrix factorization for polyphonic pitch transcription. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 109–112. IEEE, 2008.
- [19] Rafal Zdunek and Andrzej Cichocki. Nonnegative matrix factorization with quadratic programming. *Neurocomputing*, 71(10):2309–2320, 2008.
- [20] Ruohua Zhou. *Feature extraction of musical content for automatic music transcription*. PhD thesis, EPFL, 2006.

TRAINING PHONEME MODELS FOR SINGING WITH “SONGIFIED” SPEECH DATA

Anna M. Kruspe

Fraunhofer IDMT, Ilmenau, Germany

kpe@idmt.fraunhofer.de

ABSTRACT

Speech recognition in singing is a task that has not been widely researched so far. Singing possesses several characteristics that differentiate it from speech. Therefore, algorithms and models that were developed for speech usually perform worse on singing.

One of the bottlenecks in many algorithms is the recognition of phonemes in singing. We noticed that this recognition step can be improved when using singing data in model training, but to our knowledge, there are no large datasets of singing data annotated with phonemes. However, such data does exist for speech.

We therefore propose to make phoneme recognition models more robust for singing by training them on speech data that has artificially been made more “song-like”. We test two main modifications on speech data: Time stretching and pitch shifting. Artificial vibrato is also tested. We then evaluate models trained on different combinations of these modified speech recordings. The utilized modeling algorithms are Neural Networks and Deep Belief Networks.

1. INTRODUCTION

Automatic speech recognition has been a field of research for more than 30 years now and encompasses a large variety of research topics. However, speech recognition algorithms have so far only rarely been adapted to singing. One of the reasons for this seems to be that most of these tasks get harder when using singing because singing data has different characteristics, which are also often more varied than in pure speech [13]. For example, the typical fundamental frequency for women in speech is between 165 and 200 Hz, while in singing it can reach more than 1000 Hz. Other differences include harmonics, durations, pronunciation, and vibrato.

Speech recognition in singing has many interesting practical applications, such as automatic lyrics-to-music alignment, keyword spotting in songs, language identification of musical pieces or even full lyrics transcription.

A first step in many of these tasks is the recognition of

phonemes in the audio recording. We showed in [12] that phoneme recognition tends to act as a bottleneck in tasks such as language identification and keyword spotting in singing. Other publications also demonstrate that phoneme recognition on singing is more difficult than on speech [15] [6] [13]. This is further compounded by the models which have usually been trained on pure speech data.

As shown on a small scale in [6] and [12], recognition gets better when singing is used as part of the training data. The big problem with this is the lack of phoneme-annotated singing data sets.

When there is a scarcity of suitable training data, attempts are often made to generate such data artificially. For example, this is often done when models for noisy speech are required [11] [7]. In this paper, we therefore propose to make existing speech data sets more “song-like” and use these modified datasets to train models for phoneme recognition in singing. We test this procedure with the commonly used TIMIT speech dataset [10] and train Neural Networks (NNs) and Deep Belief Networks (DBNs) on modified versions of it. We then test the models’ performances on an unaccompanied singing dataset and on the test section of TIMIT.

This paper is structured as follows: We first give an introduction to the state of the art in section 2 and describe the datasets in section 3. We then present our new approach in section 4. Section 5 contains our experiments and their results. Finally, we give a conclusion in section 6 and suggest future work in section 7.

2. STATE OF THE ART

As described in [13] and in [12], there are significant differences between speech and singing data, such as pitch and harmonics, vibrato, phoneme durations and pronunciation. This makes phoneme recognition on singing harder than on speech.

Several approaches to this task have been published. In [5], Gruhne et al. describe a classical approach that employs feature extraction and various machine learning algorithms to classify singing into 15 phoneme classes. It also includes a step that removes non-harmonic components from the signal. The best result of 58% correctly classified frames is achieved with Support Vector Machine (SVM) classifiers. The approach is expanded upon in [17].

Fujihara et al. describe an approach using Probabilistic Spectral Templates to model phonemes in [4]. The pho-



© Anna M. Kruspe.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Anna M. Kruspe. “Training phoneme models for singing with “songified” speech data”, 16th International Society for Music Information Retrieval Conference, 2015.

neme models are gender-specific and only model five vowels, but also work for singing with instrumental accompaniment. The best result is 65% correctly classified frames. Mesaros presented a complex approach that is based on Hidden Markov Models which are trained on Mel-Frequency Cepstral Coefficients (MFCCs) and then adapted to singing using three phoneme classes separately [15] [14]. The approach also employs language modeling and has options for vocal separation and gender and voice adaptation. The achieved phoneme recognition rate (accuracy) on unaccompanied singing is -6.4% without adaptation and 20% with singing adaptation using 40 phonemes (the negative value is equivalent to a Levenshtein distance of 1.064, which means that there were more insertion, deletion, or substitution errors than phoneme instances). The results also improve when using gender-specific adaptation (to an average of 18.75%) and even more when language modeling is included (to 33.4%). Finally, Hansen presents a system in [6] which combines the results of two Multilayer Perceptrons (MLPs), one using MFCC features and one using TRAP (Temporal Pattern) features. Training is done with a small amount of singing data. Viterbi decoding is then performed on these posterior probabilities. On a set of 27 phonemes, this approach achieves a recall of up to 48% . It should be obvious from this overview that comparing these approaches is not easily possible. Each one uses a different dataset, a different phoneme set, and different evaluation measures.

3. DATASETS

3.1 Speech data

For training our phoneme recognition models, we used the well-known TIMIT speech dataset [10]. Its training section consists of 4620 phoneme-annotated English utterances spoken by native speakers. Each utterance is a few seconds long.

The test section of TIMIT contains similar 1680 similarly phoneme-annotated utterances. We used it to test the general performance of our models.

3.2 Singing data

To test the performance on singing data, we used the data set previously presented in [6] and [12]. It consists of the vocal tracks of 19 commercial pop songs in studio quality. We use unaccompanied singing to avoid a possible source of interference. They do not contain background music, but have been post-processed (e.g. EQ, compression, reverb). Some of them contain choir singing. Of these 19 songs, 12 were annotated with time-aligned phonemes and could therefore be used for our phoneme recognition experiments. We split these 12 songs into 562 clips, each of which roughly represents a line of the songs' lyrics.

4. PROPOSED APPROACH

An overview of our approach is shown in figure 1. We first generate five variants of the TIMIT speech dataset (training set). MFCC features are then extracted from these new datasets and used to train two models per dataset: A Neural Network and a Deep Belief Network.

Similarly, MFCCs are extracted from the TIMIT Test set and from the singing dataset. The ten previously trained models are used to recognize phonemes on these test datasets. Viterbi decoding can then be used to generate phoneme sequences. Finally, the results are evaluated.

4.1 Training data modifications

In order to make the training data more "song-like", we developed several variants of this dataset. Table 1 shows an overview over the five datasets generated from TIMIT using three modifications. Dataset N is the original TIMIT training set. For dataset P , four of the eight blocks of TIMIT were pitch-shifted. For dataset T , five blocks were time-stretched and vibrato was applied to two of them. In dataset TP , the same is done, except with additional pitch-shifting. Finally, dataset M contains a mix of these modified blocks.

In detail, the modifications were performed in the following way:

Time stretching For time stretching, we used the phase vocoder from [3], which is an implementation of the Flanagan/Dolson phase vocoder [9] [2]. This algorithm works by first performing a Short-Time Fourier Transform (STFT) on the signal and then resampling the frames to a different duration and performing the inverse Fourier transform.

As demonstrated in [12], time variations in singing are mainly performed on vowels and are often much longer than in speech. We therefore used the TIMIT annotations to only pick out the vowel segments from the utterances. They were modified randomly to a duration between 5 and 100 times the original duration and then re-inserted into the utterance. This effectively leads to more vowel frames in the training data, but since there is already a large amount of instances for each phoneme in the original training data, the effects of this imbalance should be negligible.

Pitch shifting To pitch-shift the signal, we used code from the freely available Matlab tool *AutoTune Toy* [1] which also implements a phase vocoder. In this case, the fundamental frequency is first detected automatically. The signal is then stretched or expanded to obtain the new pitch and interpolated to retain the original duration.

Using the TIMIT annotations, we split the utterance up into individual words, then generate a pitch-shifted version of each word and concatenate the results. Pitches are randomly selected from a range between 60% and 120% of the original pitch.

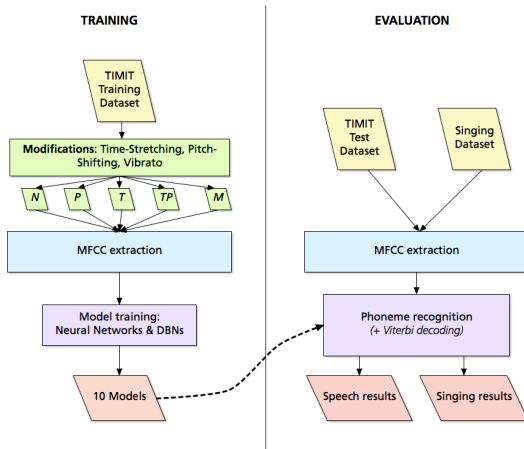


Figure 1: Overview of our phoneme recognition system

	N	P	T	TP	M
DR1	N	N	N	N	N
DR2	N	N	N	N	N
DR3	N	N	N	N	P
DR4	N	N	T	TP	TV
DR5	N	P	T	TP	TPV
DR6	N	P	T	TP	TV
DR7	N	P	TV	TPV	P
DR8	N	P	TV	TPV	TPV

Table 1: The five TIMIT variants that were used for training (rows are TIMIT blocks, columns are the five data-sets). Symbols: N - Unmodified; P - Pitch-shifted; T - Time-stretched; V - Vibrato

Vibrato The code for vibrato generation was also taken from *AutoTune Toy*. It functions by generating a sine curve and using this as the trajectory for the pitch shifting algorithm mentioned above. We used a sine of amplitude 0.2 and frequency $6Hz$. In singing, vibrato is commonly done on long sounds, which are usually vowels. Since spoken vowels are usually very short, vibrato cannot be perceived on them very well. We therefore only applied vibrato when time stretching was also applied. Vibrato was then added to the extracted and stretched vowels.

4.2 Models

Using the generated data, we trained models using two machine learning algorithms: Classical Neural Networks (NNs) and Deep Belief Networks (DBNs). Both were im-

plemented using the Theano framework for Python¹. In both cases, we first extracted Mel-Frequency Cepstral Coefficients (MFCCs) and retained the first 13 plus their deltas and double-deltas as features. We also expanded the training data to use 9 context frames. The output layer represents the 39 phonemes of the CMU Sphinx phoneme set². To make the training more exact, these phonemes were split into triphones, making the dimension of the output layer 117.

Our first models are traditional Neural Networks with two layers of 200 units each.

In recent publications, DBNs have been used very successfully for phoneme recognition (e.g. [16]). We therefore also trained DBNs on the speech data. We chose an architecture with three hidden layers and 300 units each. The first hidden layer is a Gaussian RBM.

Both models are used to generate posterior phoneme probabilities. The results for the triphone states of each phoneme are summed up into one probability for the phoneme. We then run a simple Viterbi decoding on these posteriors to generate phoneme sequences. In this decoding, all phonemes have equal transition probabilities, only the insertion penalty is variable (i.e., the transition probability to another phone). No language models are employed. We keep this post-processing simple on purpose so that the results of the various models are easily comparable.

4.3 Evaluation measures

As described in section 2, there is no single common evaluation measure for phoneme recognition in singing. We decided to compare our results using three measures:

¹ <http://www.deeplearning.org>, last checked 04/29/15

² <http://cmusphinx.sourceforge.net/>, last checked 04/29/15

Percentage of correct frames This measure describes the percentage of correctly classified frames. Correct in this case means that the exact phoneme was chosen for this frame during Viterbi decoding. A similar measure was used by Fujihara [4] and Gruhne [5].

Phoneme error rate This is the most commonly used evaluation measure in phoneme recognition for speech. It is equal to the Levenshtein distance normalized by the length of the ground truth phoneme sequence:

$$PER = \frac{D + I + S}{N} \quad (1)$$

where D are deletions, I are insertions, and S are substitutions of phonemes and N is the length of the sequence.

The accuracy measure used by Mesaros [15] [14] is the same as $1 - PER$.

Weighted phoneme error rate Mesaros also uses a measure called *correct* which ignores insertions. This makes sense if we assume that the phoneme results are used afterwards by an algorithm that is tolerant to insertions. We decided to go one step further and assume that if algorithms are tolerant to insertions, they can also be somewhat tolerant to deletions. For cases like this, Hunt suggested a weighted error rate that punishes insertions and deletions less heavily than substitutions [8]:

$$PER_{Hunt} = \frac{0.5D + 0.5I + S}{N} \quad (2)$$

5. EXPERIMENTS

We performed our experiments by training a set of models on all five TIMIT variants where all other parameters were left equal. We then classified two sets of data with these models: The unmodified “Test” part of the TIMIT speech dataset (which was not used in training) and our singing dataset. On these phoneme posterior probabilities, we ran the described simple Viterbi algorithm. The insertion penalty was optimized to generate phoneme strings who were closest in length to the ground truth phoneme strings. The three evaluation measures described in 4.3 were then calculated on the result of the Viterbi decoder.

We tested two machine learning algorithms: Neural Networks and Deep Belief Networks.

5.1 Neural Network models

Figure 2 shows the results of the Neural Network models. As figure 2a demonstrates, results for singing are generally worse than for speech. The base result for singing is a percentage of correct frames of 14.9% (model trained on the original TIMIT dataset which is denoted as N here). When comparing the models trained on the various TIMIT modifications, a slight improvement is observed for the T and M variants. For the T dataset, which includes randomly time-stretched vowels, the result improves to 15.4%. This

is a very small improvement, but it is still interesting to note. In contrast, none of the modifications improved the result on the speech data at all. The base result here is 30%. (It should be noted that much higher figures can be found in literature, but we have not yet tested improvements like language models or adaptations. Our focus for now was to compare the different TIMIT modifications).

When looking at the phoneme error rate in figure 2b instead of the pure frame accuracy, the results become more visible. The base phoneme error rate for singing is 1.16, but falls to 1.07 for the TP and M modifications. For speech, it rises from 0.6 to 0.68 (TP) and 0.66 (M) instead. The P and T modifications form a middle ground here. The P variant (randomly pitch-shifted words) does not change the results very much in either direction: It decreases the error rate on singing by 0.03 and increases it on speech by less than 0.01. The T variant (randomly time-stretched vowels) decrease the error on singing by just 0.02, but increase it on singing by 0.07.

If we weight insertions and deletions lower than modifications, the phoneme error rates decrease generally (see figure 2c). The described effects are still active when using this evaluation measure. The error rate falls from 0.88 to 0.83 on singing, and rises from 0.48 to 0.54 on speech. The tendency for P and T is similar here.

5.2 Deep Belief Network models

Figure 3 shows the same evaluation measures for the Deep Belief Networks. In general, the results are better and the effect of the various training sets is similar, but more pronounced.

The base percentage of correct frames is 14% here and rises to 19% when training on the randomly timed dataset T . On speech data, the best result is 38% for models trained on the original TIMIT data and falls for all other variants. The phoneme error rate falls from 1 to 0.91 on the singing data. Again, the results are best when the models are trained on the TP or M datasets, with the model trained on T performing just slightly worse. The lowest error rate on speech is 0.41 with the N model.

The weighted phoneme error rate sinks from 0.77 to 0.71 on the singing data.

5.3 Confusion of Deep Belief Networks

After evaluating the general performance of the Deep Belief Networks, we examined the results in detail. As an example, table 2 shows the phoneme-wise results for the singing data. The first two columns lists the frame-wise precisions and recalls when using the N model, the two columns after that show the same values for the M model, and the last column lists the three phonemes with which the concerned phoneme is confused most frequently when using the M model (except `sil`). This leads to several interesting discoveries.

It turns out that the precisions of long vowels such as `aa`, `iy`, or `oy` improve when using the M model for recognition, but some consonant accuracies become worse. This makes sense since the M modifications place an emphasis on vowels by randomly stretching them. The consonant

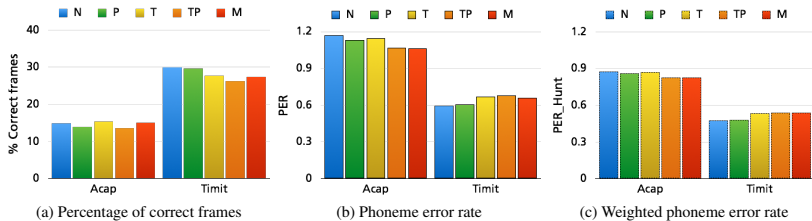


Figure 2: Evaluation measures for the results obtained with Neural Network models on singing data (Acap) and on speech data (Timit). The models were trained with the five different Timit variants (different colors).

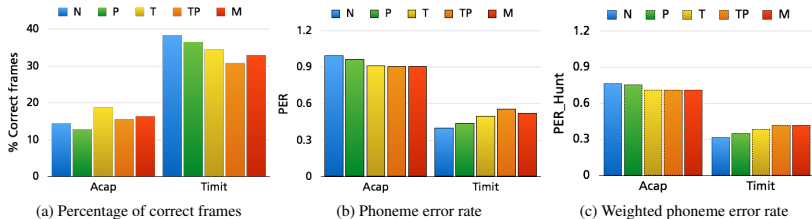


Figure 3: Evaluation measures for the results obtained with Deep Belief Network models on singing data (Acap) and on speech data (Timit). The models were trained with the five different Timit variants (different colors).

results may become worse because the training data also contains randomly pitch-shifted versions, which may not be a natural modification (this can be verified by looking at the results of the *T* model). Consonants generally seem harder to recognize since they are shorter than vowels and, for the most part, are less static over their duration. Some very bad consonant results can also be explained because they occur very rarely in the singing dataset (e.g. *zh*, *oy*). The most frequent confusion for most phonemes occurs with the *sil* state, which serves as the general “non-phoneme” state. This confusion is not displayed here.

Consonants are often confused with similar consonants (e.g. *m/n*) or with softer consonants that can be extended over a longer duration (e.g. *s*, *f*). This may be related to singing technique. However, they are also frequently confused with some vowels, particularly *uh* and *iy*. This might be caused by slight timing inaccuracies in the training annotations which become exaggerated by the time-stretching, or even by some merging of neighboring phonemes by the speaker.

Longer vowels are almost exclusively confused with other long vowels. This poses a contrast to speech, where they are usually confused with similar short vowels (e.g. *aa* → *ah*).

This becomes more conclusive when considering the confusions of short vowels. In the singing data, short vowels are very often confused with similar long vowels (e.g. *eh* → *ae*). When stretching out such short vowels in singing, singers will automatically change to such a longer vowel. Additionally, some vowels are confused with more “open” vowels (e.g. *ey* → *ae*). This is also caused by singing technique. These two very interesting effects could be ex-

ploited to improve phoneme recognition on singing in the future.

6. CONCLUSION

In this paper, we evaluated phoneme models trained on various artificially “songified” variants of the TIMIT speech dataset. The reason for this is the lack of phoneme-annotated singing datasets. We generated five such variants by randomly time-stretching vowels, randomly pitch-shifting words, and by adding vibrato to long vowels.

MFCC features were extracted from these datasets and then used to train two models each: A Neural Network and a Deep Belief Network. We then used these models to recognize phonemes in singing data and in unrelated speech data. No additional mechanics were used to improve the results, such as language modeling or gender or speaker adaptation.

In general, the results are not as good as the state of the art for the speech data. For the singing data, it is very hard to compare the results to the state of the art because other publications use different datasets, phoneme sets, and evaluation measures. However, this was not necessarily our goal - we were mainly interested in the comparative performance of the various models.

As expected, recognizing phonemes in speech seems to be much easier than in singing. Deep Belief Models performed better than their Neural Network counterparts in all test cases. For speech, the models trained on the unmodified TIMIT dataset always performed best. The best result is 38% correctly classified frames, a phoneme error rate of 0.41, and a weighted phoneme error rate of 0.32.

For singing, the models trained on the modified TIMIT da-

Ph.	Prec. <i>N</i>	Rec. <i>N</i>	Prec. <i>M</i>	Rec. <i>M</i>	Conf. <i>M</i>
aa	0.18	0.08	0.35	0.09	ao, ay, ow
iy	0.33	0.27	0.43	0.25	ey, uh, ae
ch	0.0	0.02	0.01	0.03	s, sh, iy
zh	0.26	0.02	0.0	0.0	iy, y, sh
eh	0.06	0.11	0.13	0.15	ae, ey, aa
ah	0.0	0.35	0.03	0.23	aa, er, ae
ao	0.39	0.17	0.25	0.14	aa, ow, l
ih	0.01	0.08	0.05	0.11	ey, iy, ae
ey	0.36	0.17	0.26	0.2	iy, ae, ay
aw	0.1	0.07	0.08	0.09	aa, ae, ay
ay	0.27	0.44	0.23	0.44	aa, ae, iy
ae	0.38	0.16	0.4	0.16	aa, ay, aw
er	0.22	0.1	0.19	0.08	aa, ae, eh
ng	0.07	0.16	0.06	0.11	n, uh, iy
sh	0.58	0.05	0.51	0.09	s, jh, z
th	0.0	0.0	0.0	0.0	dh, er, ey
oy	0.0	0.0	0.0	0.0	ao, ay, aa
dh	0.04	0.14	0.04	0.08	er, iy, uh
ow	0.07	0.23	0.16	0.25	ae, ae, aa
hh	0.14	0.09	0.09	0.15	iy, uh, sh
jh	0.07	0.08	0.12	0.07	y, z, sh
b	0.13	0.19	0.09	0.24	ey, m, iy
d	0.02	0.19	0.02	0.1	iy, n, er
g	0.04	0.36	0.03	0.32	y, ow, n
f	0.02	0.13	0.02	0.5	s, er, iy
k	0.02	0.37	0.05	0.31	iy, y, uh
m	0.26	0.24	0.13	0.22	uh, n, er
l	0.1	0.14	0.11	0.15	ao, er, ow
n	0.18	0.28	0.2	0.25	uh, er, uw
uh	0.23	0.02	0.15	0.02	er, ih, eh
p	0.01	0.15	0.01	0.1	er, iy, l
s	0.41	0.41	0.44	0.46	z, iy, er
r	0.16	0.24	0.17	0.18	er, aa, ao
t	0.0	0.29	0.0	0.42	s, sh, iy
w	0.24	0.26	0.19	0.2	ao, l, uw
v	0.0	0.02	0.0	0.25	er, aa, m
y	0.31	0.08	0.16	0.12	iy, y, uh
z	0.28	0.18	0.28	0.17	s, iy, n
uw	0.13	0.35	0.12	0.33	uh, er, iy

Table 2: Results per phoneme (singing data): Precision and recall with the *N* and *M* models, and most frequent confusions with *M* model (except *sil*)

taset produced better results. The best result is for singing is 18% correctly classified frames, a phoneme error rate of 0.91, and a weighted phoneme error rate of 0.71. The improvement over the models trained on the unmodified TIMIT data is 6% for the correctly classified frames, 0.09 for the phoneme error rate, and 0.06 for the weighted phoneme error rate.

The models trained on data that was only pitch-shifted only showed a very slight difference when compared to the original data. MFCCs are supposed to be pitch-invariant, and pitch-shifting therefore does not seem to make a big difference. This modification might be useful when using

other features, though. A bigger improvement on singing data was achieved when training the models on time-stretched speech data. In fact, this dataset generated the highest percentage of correctly classified frames. In this time-stretched dataset, we also applied vibrato to the stretched vowels, which happens naturally in singing. However, since the effect of pitch-shifting seemed to be small, we assume that vibrato did not have a big effect either.

There were also two datasets where both modifications (time-stretching and pitch-shifting) were mixed. Both produced the best phoneme error rates in singing.

The results were also analyzed on a phoneme-wise basis. It turned out that vowels were recognized more exactly with the modified models, while consonants were recognized somewhat worse. This may be caused by the emphasis of the generated data on longer vowels.

The most interesting effect seen in the confusion matrices is the confusion of short vowels with similar longer vowels. This has a foundation in singing technique and would be interesting to further explore to improve phoneme recognition in singing.

In general, we showed that phoneme recognition in singing can be improved when training models on artificial singing data. This finding can now be used to improve other approaches. For example, it can be combined with the techniques described in [15].

7. FUTURE WORK

As described in section 5.3, many phoneme confusions may arise from inexact or unnatural time stretching on the speech recordings. A more natural approach to this is required and we need to make sure that stretched vowels do not “leak” into neighboring consonants. We also noticed that short vowels in singing often shift towards their long versions. We will exploit this interesting effect in future phoneme recognition approaches, e.g. by allowing these confusions or composing vowels of several states.

In this paper, we tried to apply three characteristics of singing to speech recordings, but there are more, such as different pronunciations and different forming of sounds. Such other characteristics could also be tested in a similar way.

Conversely, we could also attempt to make our features and models more robust to these variations. In the past, this has often been done by adapting models trained on speech to singing in some way (also see section 2). Adaptations to gender or voice also proved helpful.

We kept the approach fairly simple for now, but the results could be improved by employing language modeling in the recognition process. We will implement this in future versions.

A possible alternative would be creating a dataset from polyphonic music data by using the lyrics and force-aligning them.

Finally, it will be interesting to see how the results of this phoneme recognition approach can be applied to practical tasks, such as lyrics-to-music alignment, keyword spotting, and language identification. For these purposes, the algorithm must also be tested on accompanied singing data.

8. REFERENCES

- [1] C. Arft. AutoTune Toy, 2010. Web resource, Last checked: 4/29/15.
- [2] M. Dolson. The phase vocoder: A tutorial. *Computer Music Journal*, 10(4):14–27, 1986.
- [3] D. P. W. Ellis. A phase vocoder in Matlab, 2002. Web resource, Last checked: 04/29/15.
- [4] H. Fujihara, M. Goto, and H. G. Okuno. A novel framework for recognizing phonemes of singing voice in polyphonic music. In *WASPAA*, pages 17–20. IEEE, 2009.
- [5] M. Grühne, K. Schmidt, and C. Dittmar. Phoneme recognition on popular music. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, Vienna, Austria, September 2007.
- [6] J. K. Hansen. Recognition of phonemes in a-cappella recordings using temporal patterns and mel frequency cepstral coefficients. In *9th Sound and Music Computing Conference (SMC)*, pages 494–499, Copenhagen, Denmark, 2012.
- [7] H.-G. Hirsch and D. Pearce. The Aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions. In *ISCA ITRW ASR*, pages 29–32, 2000.
- [8] M. J. Hunt. Figures of merit for assessing connected-word recognisers. *Speech Communication*, 9(4):329–336, 1990.
- [9] R. M. Golden J. L. Flanagan. Phase vocoder. *Bell System Technical Journal*, pages 1493–1509, November 1966.
- [10] J. S. Garofolo et al. TIMIT Acoustic-Phonetic Continuous Speech Corpus. Technical report, Linguistic Data Consortium, Philadelphia, 1993.
- [11] C. Jankowski, A. Kalyanswamy, S. Basson, and J. Spitz. NTIMIT: A phonetically balanced, continuous speech telephone bandwidth speech database. *ICASSP*, pages 109–112, 1990.
- [12] A. M. Kruspe. Keyword spotting in a-capella singing. In *15th International Conference on Music Information Retrieval (ISMIR)*, Taipei, Taiwan, 2014.
- [13] A. Loscos, P. Cano, and J. Bonada. Low-delay singing voice alignment to text. In *Proceedings of the ICMC*, 1999.
- [14] A. Mesaros and T. Virtanen. Automatic recognition of lyrics in singing. *EURASIP J. Audio, Speech and Music Processing*, 2010, 2010.
- [15] A. Mesaros and T. Virtanen. Recognition of phonemes and words in singing. In *ICASSP*, pages 2146–2149. IEEE, 2010.
- [16] A.-R. Mohamed, G. E. Dahl, and G. Hinton. Acoustic modeling using deep belief networks. *IEEE Trans. Audio, Speech, Lang. Process.*, pages 14–22, 2012.
- [17] G. Szepannek, M. Grühne, B. Bischl, S. Krey, T. Harczos, F. Klefenz, C. Dittmar, , and C. Weihs. *Classification as a tool for research*, chapter Perceptually Based Phoneme Recognition in Popular Music. Springer, Heidelberg, 2010.

GRAPH-BASED RHYTHM INTERPRETATION

Rong Jin

Indiana University
School of Informatics and Computing
rongjin@indiana.edu

Christopher Raphael

Indiana University
School of Informatics and Computing
craphael@indiana.edu

ABSTRACT

We present a system that interprets the notated rhythm obtained from optical music recognition (OMR). Our approach represents the notes and rests in a system measure as the vertices of a graph. We connect the graph by adding voice edges and coincidence edges between pairs of vertices, while the rhythmic interpretation follows simply from the connected graph. The graph identification problem is cast as an optimization where each potential edge is scored according to its plausibility. We seek the optimally scoring graph where the score is represented as a sum of edge scores. Experiments were performed on about 60 score pages showing that our system can handle difficult rhythmic situations including multiple voices, voices that merge and split, voices spanning two staves, and missing tuplets.

1. INTRODUCTION

Past decades have seen a number of efforts on the problem of Optical Music Recognition (OMR) with overviews of the history and current state of the art found at [2, 3, 8, 14]. OMR can be divided into two subproblems: identifying the music symbols on the page and interpreting these symbols, with most efforts devoted to the former problem [7, 13, 16]. However, the interpretation problem is also important for generating meaningful symbolic representations. In this paper, we focus on the rhythm interpretation of musical symbols, which appears to be the most challenging interpretation problem.

Many OMR systems [11] perform some sort of rhythm interpretation in order to play back and verify the recognized music symbols. When there are not enough notes or too many notes to match the meter of the measure, the OMR system often “flags” the measure to suggest that there is something wrong, alerting the user to correct the measure. In this way, rhythm interpretation is used as a checking tool for correcting recognized scores.

There are a few research efforts that correct recognition results automatically. Droettboom [6] proposed metric correction as part of an OMR system. Using the fact



Figure 1. Three system measures from Rachmaninoff Piano Concerto No.2 showing some of the difficulties in interpreting rhythm. All three measures are in 4/4 time.



Figure 2. Two system measures from Rachmaninoff Piano Concerto No.2 showing some of the difficulties in interpreting rhythm. Both are in 4/4 time.

© Rong Jin, Christopher Raphael.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Rong Jin, Christopher Raphael, “Graph-Based Rhythm Interpretation”, 16th International Society for Music Information Retrieval Conference, 2015.

that rhythmically coincident notes are usually aligned vertically, this work applies different corrections on inconsistent notes. Church [5] proposed a rhythmic correction with a probabilistic model that converts the rhythm of a suspicious measure to the most similar measure in the piece. Byrd [4] proposed improving OMR with multiple recognizers and sequence alignment.

The approaches mentioned above work for simpler situations such as monophonic music or measures without complex tuplets. However, some music scores, especially those for piano, are filled with rhythmically challenging situations such as missing tuplets or voices that come and go within a measure. Simple approaches are likely to fail on a significant proportion of these measures.

Our paper differs from other work we know by addressing the most challenging examples using *complete* information (the system measure), instead of trying to correct the misrecognized symbols. Our research questions are: given perfect symbol recognition is the system able to understand rhythm as a human would? When there are multiple voices interwoven in one measure, can the system separate the voices? When there are implicit symbols such as omitted rests and missing tuplets, can the system still interpret correctly?

Figures 1 and 2 show some challenging examples that illustrate the problem we address. The left measure in Figure 1 shows an example using multiple voices. When multiple voices are present it is nearly impossible to interpret rhythm without identifying these voices as such. In an effort to avoid overlapping symbols, some notes in the measure that ideally should align vertically do not. The middle measure in Figure 1 shows an example of missing tuplets (tuplets are not labeled). What is most unusual, and would likely go unnoticed by anyone other than an OMR researcher, is that these beamed groups would normally be written with two beams rather than one, though the meaning is still clear. In addition, the 9-tuplet is not explicitly indicated with a numeral — a common notational convention.

The right measure in Figure 1 shows another example of missing triplet for the 3 beamed eighth notes in the first staff, as well as a quarter note plus an eighth note pair in the second staff. A further complication is that this measure is, in some sense, incomplete, as the voice in the second staff jumps onto the first staff on the second quarter and then jumps back on the third quarter. The left measure in Figure 2 demonstrates an example of special beaming of a sextuplet where the first eighth note is separate from five beamed eighth notes. The right measure in Figure 2 demonstrates an example where all four beamed groups are triplets while the voice jumps back and forth between the two beamed groups.

The examples all seem innocent until one considers the assumptions on rhythm notation that must underlie an interpretation engine. One quickly comes to see that typical *in vivo* notation contains a fair amount of “slang” that may be readily understood by a person familiar with the idiom, but is much harder for the machine. [9] has more demon-

strations of such “slang” in music scores.

In this paper we present an algorithm that is generally capable of correctly interpreting notation such as the examples we have discussed. In our presentation, Section 2 introduces our rhythm graph and optimization on the graph score. In Section 3, we present our experiments on three scores and discuss the results.

2. METHODS

2.1 Input

We first perform optical music recognition with our *Ceres* [12] OMR system taking the score image as input. The output is stored as a collection of labeled primitive symbols such as solid note head, stem, beam, flag, and etc., along with their locations. The user deletes or adds primitive symbols using an interactive interface. Editing symbols at the primitive level allows us to keep useful information such as stem direction and beaming as well as the exact primitive locations which are important for rhythm interpretation.

After this correction phase, we assemble the primitive symbols into meaningful composite symbols (chords and beamed groups). This step is done in a simple rule-based method. Each note or rest is assigned to the staff measure it belongs to.

2.2 Rhythm Graph

We form a graph of the rhythmically relevant symbols for each system measure. The set of vertices of the graph, which we denote as V , are the notes, rests, and bar lines belonging to the system measure. All vertices are given a nominal duration represented as a rational number. For example, a dotted eighth would have nominal length $3/16$, while we give the bar lines duration 0. Sometimes the actual vertex duration can differ from the nominal length, as with missing tuplets. In these cases, we need to identify which symbols are tuplet symbols in order to interpret the rhythm correctly.

Vertices can be connected by either voice or coincidence edges, as shown in Figure 3. Voice edges, which are directed, are used for symbols whose position is understood in relation to a “previous” symbol, as in virtually all monophonic music. That is, the onset time of a symbol on the “receiving” end of a voice edge is the “preceding” symbol’s onset time plus duration. Coincidence edges link vertices that share the same onset time, as indicated by their common horizontal location. Using these edges we can infer the onset time of any note or rest connected to a bar line. We denote by E the complete collection of all possible edges.

We formulate the rhythm interpretation problem as constrained optimization. Given the set of vertices, V , and possible edges, E , we seek the subset of E , E^* , and the labeling of V that maximizes

$$H = \sum_{e \in E^*} \phi(e) + \sum_{v \in V} \varphi(l(v)) \quad (1)$$

where function $\phi(e)$ represents how plausible each edges is according to the music rules, l labels vertex v as tuplet or non-tuplet, and function $\varphi(l)$ penalizes labeling vertices as tuplet so as to favor simple interpretations whenever possible. The subset E^* and labeling are constrained to construct a consistent and connected graph.

2.3 Constructing edges

We construct the graph beginning with the left bar line (which has an onset time of 0), by iteratively connecting new vertices to the current graph with voice and coincidence edges until all vertices form a single connected component. More specifically, we connect the current vertex with a voice edge to a previously visited vertex. This vertex has to be either a bar line or a vertex in the same staff measure. (Piano staves are treated as one staff because voices often move between left and right hand parts.) This new voice edge defines a unique onset for the current vertex. Then we add coincidence edges between the current vertex and all past vertices so that both have nearly the same horizontal position and have the same onset time. We may also add coincidence edges between the incoming vertex and a past vertex having a *different* onset time, leading to a conflict that must be resolved, as discussed in Section 2.4. Different combinations of edges give different onset times to the vertices.

As an edge e is introduced to the graph we score it according to its plausibility $\phi(e)$. There are different kinds of musical knowledge [15] we hope to model in computing these scores, as follows.

1. The left bar line has an onset time of 0. The right bar line has an onset time of the measure's meter. No vertices can have onset times greater than the meter.
2. The onset times must be non-decreasing in the horizontal positions of the symbols in the image. That is, if vertex A lies to the left of vertex B it cannot have an onset that is after that of vertex B.
3. A vertex has a unique onset time. Thus, if multiple paths connect a vertex to the graph they must give the same onset time.
4. Vertices connected by coincidence edges should have the same approximate horizontal position in the image. Vertices with the same horizontal image positions should have the same onset time.
5. Vertices in a beamed group note are usually connected by voice edges, while we penalize voices that exit a beamed group before it is completed.
6. Vertices connected by a voice edge usually have the same stem direction and tend to appear at similar staff height.

The first two rules above are hard constraints that *must* be followed. When they are violated our algorithm simply will not add the offending edge. The other rules can be violated for different reasons. For example, symbols

having the same onset time may not align in the image because one is moved to avoid overlap with other symbols, or because the image is skewed or rotated through the scanning process. Such violations lead to penalties of the edge scores.

2.4 Conflict Resolution by Reinterpretation

If we disregard the right bar line and construct a spanning tree from the remaining vertices we are guaranteed that every vertex can be reached through a unique path starting from the left bar line, thus giving each vertex a unique onset time. While this approach has the compelling appeal of simplicity, it would fail in any case where the nominal note length is not the correct interpretation, as with missing tuplets. Instead, we identify such cases by allowing *multiple* paths to a vertex, and thus multiple rhythmic interpretations. When the result of these multiple paths gives conflicting onset positions for a vertex we consider reinterpreting some notes in terms of missing tuplets to resolve the conflict. In such a case we treat the earlier onset time as the correct one, while reinterpreting the path leading to the later onset time. This is because the nominal length of a tuplet note is usually greater than the true length. While there are exceptions to this rule, as with duplets in triple meter, we do not treat such cases in the current work.

As an example, consider the situation in Figure 3. Here the first coincidence edge considered (dotted line in the 1st graph) does not create any conflict since both paths give the onset position of 1/4. However, the coincidence edge for the quarter note on the top staff (dotted line in the 2nd graph) gives the onset time of 1/2 while the voice edge gives the onset time of 5/8, thereby generating a conflict. Thus we must reinterpret the path giving the later onset time of 5/8 to be consistent with the onset time of 1/2. In this case the desired interpretation is that the three eighth notes form an implicit triplet, and thus have note lengths of 1/12 rather than 1/8 (bottom graph). Another example of a conflict arises when a voice edge links to the right bar line and attributes an onset time for the bar line other than its true position (which is the meter viewed as a rational number). In this case we must reinterpret the path leading to the right bar line.

When reinterpreting we must consider the path that generates the onset position in conflict — but how far backward should we go? The collection of reinterpretable vertices could spill over into multiple voices and staff measures, thus generating an intractable collection of possibilities to consider. Here we make some simplifying assumptions to keep the computation from becoming prohibitively large. First of all, recall that we consider the staff measures of a system one at a time. After a staff measure is completely analyzed and reduced to a single interpretation, we do not consider future reinterpretations of the measure. Thus reinterpretation is confined to the current staff measure (or two staves in the case of the piano). Furthermore we do not allow the reinterpretation process to generate additional inconsistencies. This rules out the reinterpretation of any vertex connecting to a measure in a previously ana-

lyzed staff measure. Even with these restrictions the computation can still be significant, as we must simultaneously consider the possibility of a number of different triplet hypotheses, thus requiring an effort that is exponential in the number of hypotheses.

One might contrast this approach with a purely top-down model-based strategy that considers every possible rhythmic labeling. Such a strategy would be our preference if computationally feasible, and, in fact, was the approach we explored in [10]. The problem is that there are, *a priori*, a large enough collection of possible labelings so that, when coupled with unknown voicing, the computation does not always prove tractable. This is why we uncover candidates for reinterpretation prompted by coincidence edges. Thus the modeling of our algorithm lies somewhere between top-down and bottom-up recognition. It is model-based, yet it relies on the data itself to prompt certain data interpretations. While not necessarily an argument in favor of our approach, this appears to be a central part of the human strategy for rhythm understanding.

We consider several cases of reinterpretation:

1. A beamed group can be reinterpreted as a beamed triplet note of simple duration ($1/2$, $1/4$, etc.), as in the left measure of Figure 2.
2. Three consecutive vertices that add up to $3/8$ could be reinterpreted as missing triplet of total length $1/4$, as in the middle measure of Figure 2. This rule can be generalized to include other kinds of triplets (quarter note or sixteenth note) and to include triplets other than 3.
3. We can *globally* reinterpret all vertices along the voice path, as in the right measure in Figure 2, meaning that all note lengths are rescaled to create the desired collective duration.

The score function $\varphi(l(v))$ in Eqn (1) penalizes the complexity of a reinterpretation, thus favoring simple interpretations whenever possible.

2.5 Dynamic Programming for Optimization

During graph construction, each time we add a new vertex into the graph we consider adding voice edges between the new vertex and all vertices already in the graph. Thus, only considering the voice edges, the number of possible graphs with n vertices would be $n!$. Since a common system measure may have more than 50 vertices, it is computationally infeasible to search the whole graph space. This situation can be improved by dynamic programming: after any new vertex has been added to the graph, if two different graphs give identical onset times for each vertex we prune the one with lower score.

The order in which the vertices are considered is important in producing a feasible search. One way would be to visit all vertices in the system measure according to their horizontal location on the image. The problem with this approach is that the constraints imposed by the right

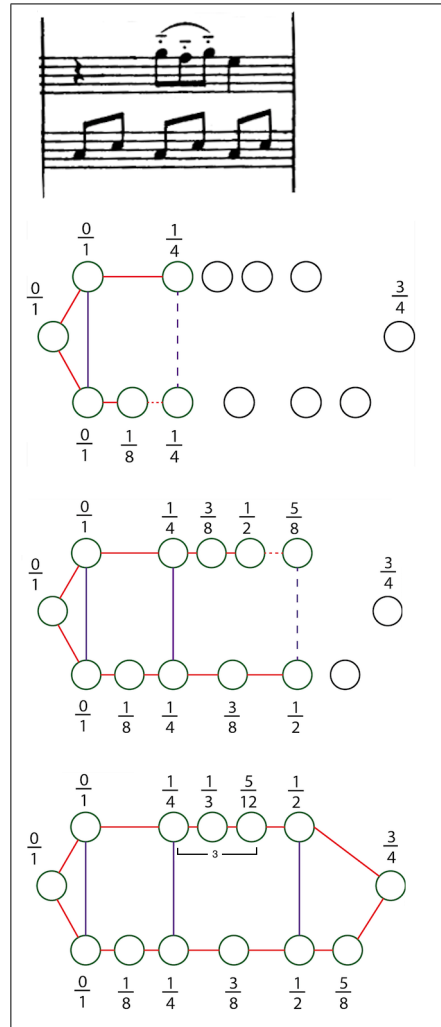


Figure 3. Constructing the rhythm graph of an example measure. Voice (red) and Coincidence (Purple) edges are automatically constructed to identify the onset time of vertices (notes, rests and bar lines).

bar line, which has a known onset position, do not come into play until the very end of the graph construction. An alternative first considers the vertices in left-right manner from a single staff measure, then continuing one-by-one with the other staff measures. Each time a staff measure is completed we continue only with the best scoring single graph. In this way, we will greatly reduce the number of partial graphs we carry through the process.

Among all measures in our experiments the maximum number of graph hypotheses we encounter during the DP computation is usually less than 100, even in the system measures with 50 to 60 vertices. The measures posing the greatest computational challenge are those having multiple voices, missing tuplets, and, at the same time, similar rhythm between the voices. The left example measure in Figure 4 shows such a case. It may seem easy for a person to recognize that there are two voices in the first staff. Here four quarter notes form one voice, and four pairs of triplets, consisting of an eighth rest and two eighth notes, form another voice. However, it's not an easy task for a computer. The second staff measure doesn't provide much information since it also has the similar missing tuplets which are hard to distinguish from nominal rhythm until one encounters the right bar line. Other measures in the same system also don't provide aligned symbols to anchor the search. The number of graph hypotheses for this system measure grows up to 2600 at the end of the measure. This measure represents the maximum number of hypotheses attained throughout our experiments. This is still easily feasible computationally.

3. EXPERIMENTS

In the experiments, we have chosen three different scores of varying degrees of rhythmic complexity for evaluation, all taken from the IMSLP [1].

3.1 Rachmaninoff Piano Concerto No.2

The orchestra score of Rachmaninoff Piano Concerto No. 2 is a highly challenging example for our rhythm interpretation algorithm. The score has 371 system measures, with each system measure containing up to 15 staff measures. The piece covers different types of rhythmic difficulties such as polyphonic voices, missing tuplets, and voices moving between staff measures. In addition some pages of the score are rotated and skewed due to the scanning process, creating difficulty detecting coincidence between notes.

We get 355 out of 371 (95.7%) system measures correctly. In the following paragraphs, we will discuss three representative examples in which our system fails to find the correct rhythm.

Failure case 1 In the left example in Figure 4 we fail to interpret all the missing triplets. The result produced by our system did not recognize the first and last triplet in the first staff, instead treating those beamed eighth notes as normal eighth notes. The system gives the left eighth note in the beam the same onset time as the eighth note

rest, explaining it as coincidence with the eighth note rest since they almost align vertically. In this case we found that the correct interpretation was actually generated by our system, but survives with a lower score. This type of scenario, where the correct interpretation survives but does not win, occurs a number of times in our experiments. In this case, the reason is because we give a high penalty for tuplet reinterpretation, while a give comparatively lower penalty when allegedly coincident symbols are not perfectly aligned. Therefore, the state that has fewer tuplets but worse alignment gets a higher score.

Failure case 2 The right example in figure 4 is another example where our system does not produce the correct rhythm. The difficulty in this measure is the voice that moves between the treble and bass staves of the piano. While we successfully recognized two missing sextuplets in the treble staff, we failed to recognize that the quarter note in treble staff and eighth note in the bass staff form a triplet. In our result, they are interpreted as a normal quarter note and a normal eighth note with the eighth note aligned to the 3rd sixteenth through a coincidence edge. This happens because we impose a penalty for interpreting a missing tuplet, while the eighth note aligns reasonably well with the third 16th note, providing a plausible explanation. However, the isolated eighth note is the only note that has the wrong onset time. This case also shows that our algorithm is capable of recovering from local errors to produce mostly correct results, even though not perfect.

Failure case 3 Our third incorrect case is shown in the left of Figure 5. In the first staff of this example, the dotted half note chord and first eighth note in the first beam group both begin at the start of the measure. However, we have a maximal horizontal distance between two notes that have the same onset time, which serves the important role of pruning graphs graphs that exceed this threshold — usually this is the correct thing to do. In this particular case these two notes exceeded the threshold, thus we lose the correct interpretation. For such a case, we can always make the threshold larger, but this weakens the power of the alignment rule elsewhere. Of course, there will always be special cases where our threshold is not large enough. In the right measure in Figure 5, the eighth rest and whole note “high” c in the first staff are very far away from the half note in staff three due to the long grace note figure. Presumably the grace note figure begins on the *beat*, so the coincidence suggested by the score is correct, though this peculiarity lies outside of the basic modeling assumptions we have employed: here two notes at the same rhythmic position are not intended to sound at the same time! We have a few other examples of this general type of failure, such as when we can't compute horizontal distances accurately due to image skew. Given the reasons above, we decide to keep the threshold as strict as it is, because it provides a significant help with keeping the computation tractable.



Figure 4. Examples for failure case 1 and failure case 2 from Rachmaninoff Piano Concerto No. 2. Both measures are in 4/4 time.



Figure 5. Examples for failure case 3 from Rachmaninoff Piano Concerto No. 2. Both measures are in 4/4 time.



Figure 6. Two examples from Debussy's 1st Arabesque. Both measures are in 4/4 time.

3.2 Borodin String Quartet No.2, 3rd Movement

We also tested on the 3rd movement (*Notturmo*) from Borodin's 2nd String Quartet. This is a "medium" difficulty score consisting of 4 staves for each system. The third movement has 180 systems measures over 6 pages. 22 out of 180 system measures contain triplets, and, while all of these are explicitly notated with numerals in the score, we deliberately didn't include these in our rhythm interpretation process. The system gets 100 percent correct rhythm on all of these measures.

3.3 Debussy 1st Arabesque

Usually the more staves in a system, the more coincidence edges between different staves, thus providing anchors for reinterpretation when needed. Thus solo piano music can be particularly challenging with only two staves. In measures that are monophonic or homophonic we can't identify inconsistencies until we reach the end of the measure as both nominal and tuplet hypotheses are consistent with spacing. In order to demonstrate that our system is also capable of handling these challenges, we experimented on the first of the two Debussy Arabesques, containing 107 measures.

This piece has a variety of rhythmic difficulties. 73/107 (68%) of the system measures have at least one, and up to six missing tuplets, while 17/107 measures contain voices moving between the two staves. This latter category is particularly difficult because the measures are monophonic as in Figure 6, and thus do not provide coincidence clues. Therefore, our algorithm only sees conflicts at the end of the measure and must reinterpret the entire measure at once. However, our results show that we are generally capable of handling such situations. There's only one measure that we don't get exactly correct as shown in the right of Figure 6. In this measure, there are four missing beamed group triplets. In our best scoring solution, we found the first and last triplets but are missing the middle two. The correct interpretation also survives into the final list but with a lower score.

4. CONCLUSION

We have presented a graph-based rhythm interpretation system. Experiments show that given the perfect symbol recognition, our system is generally capable of interpreting difficult notation involving separating multiple voices and identifying implicit symbols such as missing tuplets. It also shows that it's a difficult and interesting problem and worth further exploration. One possibility will be using trained penalty parameters for a particular score. A rare notation or rhythmic pattern could appear repeatedly in one score, thus we hope an adaptive model would improve the result. Also, since there are always exceptions in all music-related questions, human interactive methods are another interesting direction to explore.

5. REFERENCES

- [1] International music score library project. <http://imslp.org/>.
- [2] D. Bainbridge and T. Bell. The challenge of optical music recognition. *Computers and the Humanities*, 2001.
- [3] D. Blostein, H. Dorothea, and H. Baird. A critical survey of music image analysis. *Structured Document Image Analysis*. Springer Berlin Heidelberg, 1992.
- [4] D. Byrd. Prospects for improving OMR with multiple recognizers. In *Proceedings of the International Symposium on Music Information Retrieval*, 2006.
- [5] M. Church and M. Cuthbert. Improving rhythmic transcriptions via probability models applied post-OMR. In *Proceedings of the International Symposium on Music Information Retrieval*, 2014.
- [6] M. Droettboom, I. Fujinaga, and K. Macmillan. Optical music interpretation. In *Structural, Syntactic, and Statistical Pattern Recognition*, 2002.
- [7] I. Fujinaga. Adaptive optical music recognition ph.d. thesis, mcgill university,montreal. 1997.
- [8] I. Fujinaga. Optical music recognition bibliography. <http://www.music.mcgill.ca/~ich/research/omr/omrbib.html>, 2000.
- [9] J. Hook. How to perform impossible rhythms. *Society for Music Theory*, 2011.
- [10] R. Jin and C. Raphael. Interpreting rhythm in optical music recognition. In *Proceedings of the International Symposium on Music Information Retrieval*, 2012.
- [11] G. Jones, B. Ong, I. Bruno, and K. Ng. Optical music imaging: music document digitisation, recognition, evaluation, and restoration. *Interactive Multimedia Music Technologies*, pages 50–79, 2008.
- [12] C. Raphael and R. Jin. The Ceres system for optical music recognition. In *International Conference on Pattern Recognition Applications and Methods*, 2014.
- [13] C. Raphael and J. Wang. New approaches to optical music recognition. In *Proceedings of the International Symposium on Music Information Retrieval*, 2011.
- [14] A. Rebelo, G. Capela, and J. Cardoso. Optical music recognition: state-of-the-art and open issues. *International Journal of Multimedia Information Retrieval*, 2012.
- [15] G. Reed. Music notation:a manual of modern practice. 1979.
- [16] F. Rossant and I. Bloch. Robust and adaptive OMR system including fuzzy modeling,fusion of musical rules, and possible error detection. In *EURASIP Journal on Applied Signal Processing*, 2007.

LET IT BEE – TOWARDS NMF-INSPIRED AUDIO MOSAICING

Jonathan Driedger, Thomas Prätzlich, Meinard Müller

International Audio Laboratories Erlangen

{jonathan.driedger,thomas.praetzlich,meinard.mueller}@audiolabs-erlangen.de

ABSTRACT

A swarm of bees buzzing “Let it be” by the Beatles or the wind gently howling the romantic “Gute Nacht” by Schubert – these are examples of *audio mosaics* as we want to create them. Given a *target* and a *source* recording, the goal of audio mosaicing is to generate a *mosaic* recording that conveys musical aspects (like melody and rhythm) of the target, using sound components taken from the source. In this work, we propose a novel approach for automatically generating audio mosaics with the objective to preserve the source’s timbre in the mosaic. Inspired by algorithms for *non-negative matrix factorization* (NMF), our idea is to use update rules to learn an activation matrix that, when multiplied with the spectrogram of the source recording, resembles the spectrogram of the target recording. However, when applying the original NMF procedure, the resulting mosaic does not adequately reflect the source’s timbre. As our main technical contribution, we propose an extended set of update rules for the iterative learning procedure that supports the development of sparse diagonal structures in the activation matrix. We show how these structures better retain the source’s timbral characteristics in the resulting mosaic.

1. INTRODUCTION

Using the sounds in a recording of buzzing bees to recreate a recording of the song “Let it be” by the Beatles is a typical example of an audio mosaic. In this example, the recording of the bees serves as *source*, while the Beatles recording is called the *target*. Ultimately, one should be able to identify the target recording when listening to the mosaic, but at the same time perceive the timbre of the source sounds. Therefore, the audio mosaic of “Let it be” with the bee recording could give the impression of bees being musicians, buzzing the song’s tune.

Audio mosaicing is an interesting audio effect which has found its way into both artistic work as well as academic research. Artists like John Oswald used thousands of manually selected source audio snippets to create new

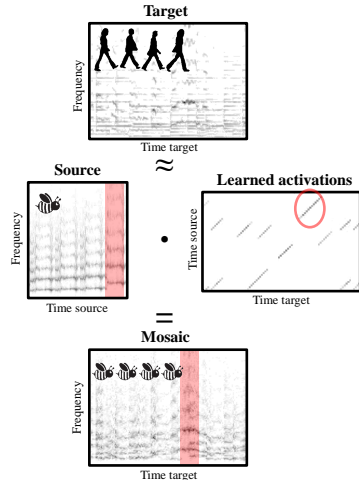


Figure 1. Schematic overview of our proposed audio mosaicing method. The sparse diagonal structures in the activation matrix are important in order to preserve the timbre of the source in the mosaic.

musical compositions¹ and real-time audio mosaicing has been used by musicians as an instrument in live performances [4, 22]. Over the years, many different systems for audio mosaicing were proposed [1, 3, 5, 11, 13, 17, 18]. The core idea of most automated systems is to split the source into short audio segments, which are suitably concatenated afterwards to match spectral and temporal characteristics of the target [19].

In this work, we propose a novel way to create audio mosaics. Our idea is to learn an *activation matrix* that, when multiplied with the spectrogram of the source recording, approximates the spectrogram of the target recording (see Figure 1). The source spectrogram hereby serves as a *template matrix* which is fixed throughout the learning process. This way, as opposed to many previous automated mosaicing approaches, a frame of the target can be re-synthesized as the superposition of several spectral frames of the source, thus allowing “polyphony” of the source sounds.

¹ Especially on his album *Plexure* [16].

As a first contribution, we propose an audio mosaicing procedure which is inspired by well-known algorithms for *non-negative matrix factorization* (NMF) [14]. Keeping the template matrix fixed (the source’s magnitude spectrogram), this basic procedure learns an activation matrix by iteratively applying a standard NMF update rule to a randomly initialized matrix. Experiments show that in case the source recording offers an appropriate amount of different sounds, this procedure can closely approximate the spectrogram of the target recording. However, the source’s timbre is often barely recognizable in the resulting mosaics. The reason is that the procedure recreates every target frame independently, thus destroying temporal characteristics of the source in the final audio mosaic. Furthermore, the method can superimpose an arbitrary number of spectral frames from the source to construct a good numerical approximation of a single target frame. A superposition of a large number of source sounds may however result in a timbre that is no longer similar to the actual timbre of the source. Therefore, an exact approximation of the target’s spectrogram cannot be our procedure’s sole goal.

As our main technical contribution, we therefore propose an extended set of update rules that supports the development of sparse diagonal structures in the activation matrix during the learning process (see the activation matrix in Figure 1). Rather than single frames, diagonal structures activate whole frame sequences in their original order. This preserves the source’s temporal characteristics in the resulting mosaic. Furthermore, the extended set of update rules also limits the number of simultaneous activations, making the learned activation matrix sparse and reducing the problem of too many source sounds being audible simultaneously. This way, we trade some approximation quality for a better preservation of the source’s timbre.

The idea of activating sequences of frames is inspired by methods like *non-negative matrix factor deconvolution* (NMF-D) and related formulations [20,21], where template sequences of frames from a dictionary are activated by single activation values. However, our approach is conceptually different. Instead of changing the NMF problem formulation, our approach stays in the standard NMF setting, supporting the activation of whole frame sequences directly in the activation matrix with additional update rules. Besides being computationally very efficient and easy to implement, this also has the advantage that we do not need to choose a maximal length of the sequences as in NMF-D. Similarly, the sparseness constraint imposed by our procedure is not enforced by penalty terms in the problem formulation (as for example in [8, 10, 12, 23]), but also by additional update rules.

The remainder of this paper is structured as follows. In Section 2 we introduce the basic concept of using NMF-inspired update rules for the task of audio mosaicing. In Section 3 we present the extended set of update rules that supports the development of sparse diagonal structures in a learned activation matrix. The effects of these update rules on the audio mosaics are discussed and demonstrated in Section 4.

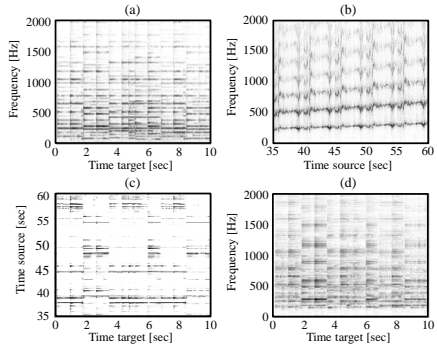


Figure 2. Basic NMF-inspired audio mosaicing. **(a):** Magnitude spectrogram of “Let it be” V (target). **(b):** Magnitude spectrogram of a recording of bees W (source). **(c):** Activation matrix H . **(d):** The product WH (mosaic).

2. BASIC NMF-INSPIRED AUDIO MOSAICING

Non-negative matrix factorization (NMF) has been applied very successfully in a large variety of music processing tasks and beyond. Given a non-negative matrix $V \in \mathbb{R}_{\geq 0}^{N \times M}$, the goal of NMF is to decompose this matrix into two factors $W \in \mathbb{R}_{\geq 0}^{N \times K}$ and $H \in \mathbb{R}_{\geq 0}^{K \times M}$, where $N, M, K \in \mathbb{N}$. The distance between the product WH and the matrix V is minimized with respect to some distance measure, for example the Kullback-Leibler divergence

$$(V \| WH) = \sum_{nm} V_{nm} \log \frac{V_{nm}}{(WH)_{nm}} - V_{nm} + (WH)_{nm}. \quad (1)$$

In the context of music processing, the matrix V is usually a magnitude spectrogram of a music recording, the matrix W is interpreted as a set of spectral templates, and the matrix H constitutes an activation matrix. Non-zero values in a row of H activate the associated template in W at the respective time instance. The two factors W and H are usually learned by iteratively applying multiplicative update rules to two suitably initialized matrices [14].

Fixing the template matrix W to be the magnitude spectrogram of the source recording, the basic idea of our proposed audio mosaicing approach is to learn only the activation matrix H . More precisely, we proceed as follows. Given the target recording x_{tar} and the source recording x_{src} , we first compute the complex valued spectrograms X_{tar} and X_{src} by applying the short-time Fourier transform (STFT) to both recordings. Afterwards, we set $V := |X_{tar}|$, $W := |X_{src}|$, and randomly initialize $H^{(1)} \in (0, 1]^{K \times M}$. Fixing a number of iterations L , we then iteratively update H with

$$H_{km}^{(\ell+1)} = H_{km}^{(\ell)} \frac{\sum_n W_{nk} V_{nm} / (WH^{(\ell)})_{nm}}{\sum_n W_{nk}}, \quad (2)$$

for $k \in [1 : K]$, $m \in [1 : M]$, and the iteration index $\ell \in [1 : L - 1]$. Finally, we set $H := H^{(L)}$. The learned activation matrix H is then multiplied with the complex valued X_{src} , yielding the complex valued spectrogram of the audio mosaic $X_{mos} := X_{src}H$. To compute the audio mosaic x_{mos} , we apply an “inverse” STFT to the spectrogram X_{mos} , which also adjusts the phases such that artifacts from phase discontinuities are reduced [9].

Figure 2 shows this basic procedure applied to our running example. In Figure 2a we see an excerpt of the magnitude spectrogram of the song “Let it be”. Our goal is to create an audio mosaic of this song, using the recording of buzzing bees, which can be seen in Figure 2b. To increase the range of different pitches occurring in our source, we used a pitch-shifting algorithm [6] to create differently pitched versions of the bee recording and concatenated them. Figure 2c shows an excerpt of the activation matrix H , derived by applying the basic procedure described above. A first observation about H is the predominance of horizontal activation structures. These patterns correspond to single spectral frames in the source which are activated repeatedly to mimic the stable spectral structures in the target. Although the resulting mosaic, shown in Figure 2d, closely resembles these spectral structures, one can hear a “stuttering” effect when listening to the reconstructed audio recording. This stuttering originates from the same frame of the source being repeated over and over again. In Section 3.1, we aim to prevent the learning process from activating the same frame in fast repetition with an additional update rule.

A second observation is that the matrix H usually activates many source frames simultaneously. The learning process can thus closely approximate the spectral shapes of the target frames. However, in the context of audio mosaicing, this has several drawbacks. Since H is multiplied with the complex spectrogram X_{src} , phase cancellation artifacts may arise when superimposing many complex spectral frames. This way, especially low pitched sounds tend to cancel each other out and are not audible in the final audio mosaic. Furthermore, since a sound’s timbre is also closely related to the energy distribution in its frequency spectrum, adapting the spectral shapes may change the timbre of the source. An update rule which sets a limit on the maximal number of simultaneous activations is presented in Section 3.2.

A third problem connected with the activation matrix shown in Figure 2c is the loss of temporal characteristics of the source. The typical “buzzing sound” of the bees, which results from pitch modulations (see Figure 2b), is lost in the mosaic (see Figure 2d). This is the case since the spectral frames of the source are activated independently of their order in the source spectrogram. To preserve some temporal characteristics, the update rule presented in Section 3.3 supports the development of diagonal structures in the activation matrix.

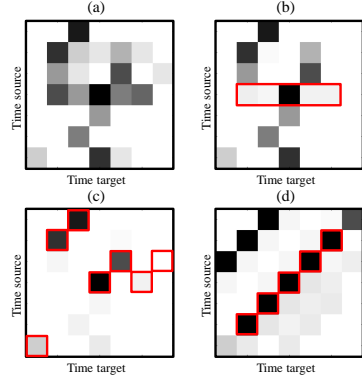


Figure 3. (a): Activation matrix $H^{(\ell)}$. (b): Repetition restricted activation matrix $R^{(\ell)}$. The horizontal neighborhood is indicated in red. (c): Polyphony restricted activation matrix $P^{(\ell)}$. For each column, the highest value is indicated in red. (d): Continuity enhancing activation matrix $C^{(\ell)}$. The diagonal kernel is indicated in red.

3. LEARNING SPARSE DIAGONAL ACTIVATIONS

The core idea to overcome the issues of the basic NMF-inspired audio mosaicing procedure is to impose specific constraints on the learned activation matrices by adapting the iterative update process. As discussed in the previous section, we identified three main problems of the mosaics generated by the basic procedure, all related to properties of the derived activation matrices. First, horizontal activation patterns cause stuttering artifacts in the mosaics. Second, too many simultaneous activations lead to phase cancellations and overfitting of the spectral shapes. Third, the source’s temporal characteristics are destroyed by activating source frames independently of each other. We therefore introduce additional update rules to approach these issues, see also Figure 3.

3.1 Avoiding repeated activations

To avoid activating the same spectral frame of the source in subsequent time-instances, the idea is to only keep the highest activations in a horizontal neighborhood of the matrix H , suppressing the remaining values. However, we do not want to interfere too much with the actual learning process in the first few update iterations. The amount of suppression applied to the smaller values is therefore dependent on the iteration index ℓ . Given the activation matrix $H^{(\ell)}$, the size of a horizontal neighborhood r , and the number of iterations L , we compute a *repetition restricted* activation matrix $R^{(\ell)}$ by

$$R_{km}^{(\ell)} = \begin{cases} H_{km}^{(\ell)} \mu_{km}^{r,(\ell)} & \text{if } H_{km}^{(\ell)} = \mu_{km}^{r,(\ell)} \\ H_{km}^{(\ell)} (1 - \frac{(\ell+1)}{L}) & \text{otherwise} \end{cases}, \quad (3)$$

with $\ell \in [1 : L - 1]$ and $\mu_{km}^{r,(\ell)}$ being the maximum value of $H^{(\ell)}$ in a horizontal neighborhood

$$\mu_{km}^{r,(\ell)} = \max(H_{k(m-r)}^{(\ell)}, \dots, H_{k(m+r)}^{(\ell)}) . \quad (4)$$

Note that the suppression of smaller values becomes strict in the last update iteration for $\ell = L - 1$. Intuitively, the parameter r defines the minimal horizontal distance (and therefore the minimal time interval) between two activations of the same source frame. Figure 3b shows the repetition restricted activation matrix $R^{(\ell)}$ derived from the toy example activation matrix shown in Figure 3a, using $r = 2$, $\ell = 8$, and $L = 10$. As opposed to $H^{(\ell)}$, there are no two dominant values next to each other in $R^{(\ell)}$.

3.2 Restricting the number of simultaneous activations

Next, we address the problem of too many simultaneous activations. Setting a limit $p \in \mathbb{N}$ on the number of activations in one column of the activation matrix, we compute a *polyphony restricted* activation matrix $P^{(\ell)}$ in a similar manner as $R^{(\ell)}$ by

$$P_{km}^{(\ell)} = \begin{cases} R_{km}^{(\ell)} & \text{if } k \in \Omega_m^{p,(\ell)} \\ R_{km}^{(\ell)} (1 - \frac{\ell+1}{L}) & \text{otherwise} \end{cases} , \quad (5)$$

where $\Omega_m^{p,(\ell)}$ contains the indices of the p highest values in the m^{th} column of $R^{(\ell)}$. The parameter p can be directly interpreted as the desired degree of polyphony in the mosaic. For example, setting $p = 1$ results in a mosaic where the source sounds are not heavily superimposed but mainly concatenated to mimic the most dominant features of the target. In Figure 3c, we see the polyphony restricted activation matrix $P^{(\ell)}$ derived from $R^{(\ell)}$, using $p = 1$. One can see that in $P^{(\ell)}$ there is (at most) one single dominant value left in every column.

3.3 Supporting time-continuous activations

To support the development of diagonal structures that activate successive frames of the source, we now compute a *continuity enhancing* activation matrix $C^{(\ell)}$. The idea here is to convolve the matrix P with a diagonal kernel. Choosing $c \in \mathbb{N}$, which defines the length of the kernel, we compute

$$C_{km}^{(\ell)} = \sum_{i=-c}^c P_{(k+i)(m+i)}^{(\ell)} . \quad (6)$$

Intuitively, the length $2c + 1$ of the kernel defines the minimal number of source frames that we would like to successively activate. Figure 3d shows the matrix $C^{(\ell)}$ for our toy example, computed with $c = 2$. Note that in $C^{(\ell)}$ the number of simultaneous dominant activations may locally exceed the limit which was imposed in the computation of the polyphony restricted activation matrix $P^{(\ell)}$. In practice, this is however not a problem and even desirable since this way, the diagonal structures can overlap with each other to some degree. Therefore, the corresponding audio segments of the source are overlapped in the final mosaic as well, leading to smooth transitions between them.

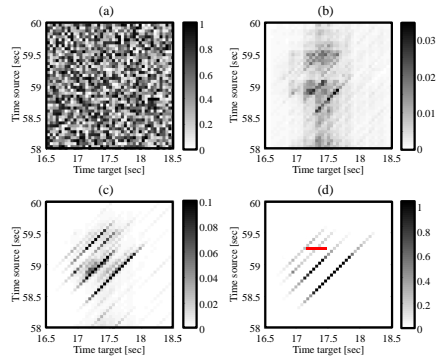


Figure 4. The activation matrix H for the mosaic of “Let it be” with a recording of bees in different states. **(a):** $H^{(1)}$. **(b):** $H^{(3)}$. **(c):** $H^{(6)}$. **(d):** $H^{(10)}$. The repetition restricting neighborhood is indicated in red.

3.4 Adapting the activations to fit the target

Finally, we perform the standard NMF update step to let the mosaic adapt to the target again. Similarly to Equation (2), we compute the activation matrix for the next iteration by

$$H_{km}^{(\ell+1)} = C_{km}^{(\ell)} \frac{\sum_n W_{nk} V_{nm} / (W C^{(\ell)})_{nm}}{\sum_n W_{nk}} . \quad (7)$$

In summary, a single update step of the activation matrix H is computed by applying Equations (3), (5), (6), and (7) sequentially.

Note that in one update iteration, the three intermediate update rules (3), (5), and (6) are insensitive to the target and therefore may increase the distance measure of Equation (1). However, as already discussed in Section 1, we are not interested in minimizing this measure, but trade some approximation accuracy for a better preservation of the source’s timbre. In practice, our procedure usually yields an activation matrix that, when multiplied with the source spectrogram, approximates the target spectrogram to a sufficient degree, while preserving the source’s timbre in the mosaic much better than the basic procedure described in Section 2.

Figure 4 shows an excerpt of the activation matrix H of our running example “Let it be” for several iteration indices ℓ . Here, we set the repetition restriction parameter to $r = 3$, the limit of simultaneous activations to $p = 10$, the kernel parameter to $c = 3$ (resulting in a diagonal kernel of length 7), and the number of update iterations to $L = 10$. Figure 4a shows the random initialization of the activation matrix $H^{(1)}$. After two iterations, one can already notice diagonal patterns in $H^{(3)}$, see Figure 4b. Figure 4c shows the activations after another three update iterations. The diagonal patterns in $H^{(6)}$ are even more prominent and one can observe that separate diagonal structures start to

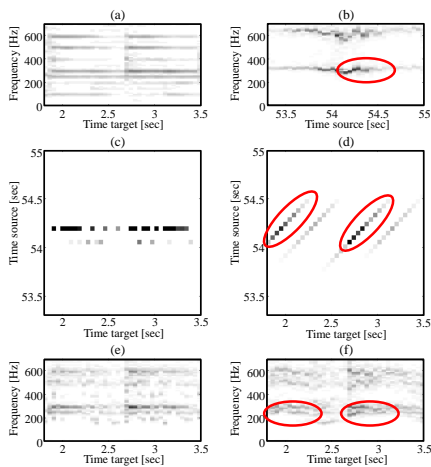


Figure 5. The effect of diagonal activation patterns. (a): Spectrogram of the target recording “Let it be”. (b): Spectrogram of the source recording of buzzing bees. (c): Activation matrix H derived with the basic approach. (d): Activation matrix H derived with the extended set of update rules. (e): Spectrogram of the audio mosaic resulting from the basic approach. (f): Spectrogram of the audio mosaic resulting from the extended procedure.

emerge, leaving regions of lower values inbetween them. In Figure 4d, the activation matrix $H^{(10)}$ is shown. In this final activation matrix, four clear diagonal structures have emerged. The remaining activations are outside the visible range. Looking at the two upper diagonals, one can see that although they seem to be rather close together, they obey the repetition restricting horizontal neighborhood indicated in red. Furthermore, it is noteworthy that the length of the diagonals greatly exceeds the length of the diagonal kernel. For example, while we used a diagonal kernel of length 7, the lowest diagonal has a length of 25 non-zero activations, corresponding to an audio segment in the source of roughly one second. This means that the procedure uses a whole one-second patch of source audio material to recreate the target between second 17 and 18.

4. EXPERIMENTS AND EXAMPLES

In this section, we both visually and acoustically demonstrate the effectiveness of our proposed method. As discussed in previous sections, the main drawbacks of the basic audio mosaicing approach described in Section 2 were both the loss of temporal characteristics and spectral shapes of the source sounds in the resulting audio mosaics. The idea was to approach these problems by supporting the development of sparse diagonal structures in the activation matrix with an extended set of update rules. In the follow-

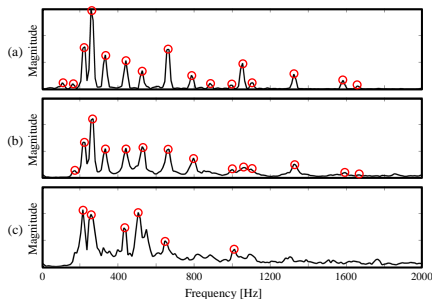


Figure 6. Comparison of spectral shapes. (a): A single spectral frame of the target recording (“Let it be”). Harmonics are indicated by red circles. (b): The spectral frame of the mosaic computed with the basic procedure at the same temporal position. Harmonics which are present in both the original frame as well as in the mosaic are indicated by red circles. (c): The spectral frame of the mosaic computed by using the extended set of update rules.

ing, we exemplify how these structures can preserve the source’s desired characteristics in the audio mosaic.

4.1 Preserving temporal characteristics of the source

In Figure 5, we once again revert to our running example. Here, spectrogram excerpts of the target recording “Let it be” as well as the source recording of buzzing bees are shown in Figures 5a and 5b, respectively. The spectrogram of the target recording exhibits sounds with very stable pitches, resulting from the solo piano at the beginning of the song. In contrast, the buzzing of the bees leads to rather strong amplitude modulations that are characteristic for the sound. Figure 5c shows an excerpt of the activation matrix H as derived by the basic NMF-inspired audio mosaicing procedure. In this excerpt of H , only two different spectral frames of the source are activated repeatedly by the procedure to mimic the stable pitch of the piano sound. The resulting spectrogram of the audio mosaic, shown in Figure 5e, approximates the target’s spectrogram quite precisely. However, the characteristic pitch modulations of the buzzing bee sound are lost almost completely. Looking at Figure 5d, one can see the activation matrix H derived by our proposed procedure based on the extended set of update rules. The diagonal patterns shown activate segments of the source that have a duration of roughly half a second. As can be seen by comparing the regions marked in red in the source (Figure 5b) and the mosaic spectrogram (Figure 5f), the temporal structures of these segments are preserved in the mosaic. While the mosaic computed with the extended set of update rules exhibits a lot of pitch modulations, which reflect the preserved timbre of the buzzing bee sound, the tonal content as well as rhythmic structures of the target are still maintained. For example, the two strong partials of the target recording at around 270 Hz and

Name of the target	Description of the target	Name of the source	Description of the source
LetItBe	An excerpt of the song "Let it be" by the Beatles (piano & singing).	Bees	Recording of a buzzing swarm of bees.
GuteNacht	An excerpt of "Gute Nacht" by Franz Schubert which is part of the romantic <i>Winterreise</i> song cycle, taken from [15].	Wind	Recording of howling wind.
FunkJazz	An excerpt from a jazz piece performed by the band "Musie Delta" (saxophone, synthesizer, bass, and drums), taken from [2].	Whales	Recording of whale songs and whale sounds.
Stepdad	Excerpt from the song "My leather, my fur, my nails" by the pop band Stepdad (synthesizers, drums, and singing).	Chainsaw	Recording of a chainsaw's sawing and engine sounds.
Freischütz	Excerpt from the opera "Der Freischütz" by Carl Maria von Weber (full orchestra, applause at the end).	AirRaid	Recording of an air raid siren.
Vermont	An excerpt of the song "Vermont" by the band "The Districts" (singing, guitar, bass, and drums), taken from [2].	RaceCars	Recording of engine sounds of starting race cars.

Table 1. List of target and source recordings used in our experiments.

300 Hz in Figure 5a are also visible in the audio mosaic in Figure 5f, only this time pitch modulated. Similarly, the onset in the target at second 2.6 is present in the mosaic as well.

4.2 Preserving spectral shapes of the source

In Figure 6, we investigate typical spectral shapes of the target as well as the mosaic for our running example. Figure 6a shows the spectral frame of the target's spectrogram at second 4.6 as a frequency-magnitude plot. One can see the harmonic structure with several clear partials in this frame, resulting from the piano sound in the target. The corresponding spectral frame of the mosaic computed by the basic procedure shown in Figure 6b shows a very similar spectral structure. Most of the harmonics visible in the target are also present in this frame (indicated by the red circles) and even the relations between peak heights are often preserved. In contrast, the spectral frame of the mosaic computed with the extended set of update rules only roughly corresponds to the spectral shape of the target frame, see Figure 6c. However, some of the dominant peaks in the target frame are still present in the mosaic, leading to a sound that captures only the dominant tonal characteristics of the target. The noisy timbre of the buzzing bees, visible by the increased noise level in the frame, is therefore preserved.

4.3 Audio examples

In order to also give an auditory demonstration of our method, we set up an accompanying website for this paper at [7]. On this website, one finds the target recordings as well as source recordings listed in Table 1. To ensure that each source recording offers an adequate pitch range, we computed several pitch-shifted versions of it (using a pitch-shifting algorithm from [6]) and concatenated them. For each pair of target and source, we then generated an audio mosaic using both the basic mosaicing procedure described in Section 2 as well as the procedure based on the extended set of update rules proposed in Section 3. For these experiments, we used music recordings sampled at 22050 Hz, an STFT frame length of 2048 samples and a hop size of 1024 samples to compute the spectrograms. In order to derive the activation matrices for both procedures, we performed $L = 20$ iterations of the respective

update steps. For the extended set of update rules, we set the repetition restriction parameter to $r = 3$, the limit of simultaneous activations to $p = 10$, and the kernel parameter to $c = 3$. To reconstruct time-domain signals from the derived complex valued mosaic spectrograms, we finally performed 20 iterations of the STFT inversion procedure proposed in [9].

5. CONCLUSION AND FUTURE WORK

In this work we presented a novel approach for automatically generating an audio mosaic of a target recording using the sounds from a source recording. The core idea of this NMF-inspired procedure was to learn an activation matrix that, when multiplied with the spectrogram of the source recording, yields the spectrogram of the mosaic recording. As our main technical contribution, we proposed an extended set of update rules that supports the development of sparse diagonal structures in the activation matrix during the learning process. Our experiments showed that these diagonal activation structures correspond to the activation of whole sequences of spectral frames and help to preserve timbral characteristics of the source in the mosaic.

In future work we want to investigate if our proposed procedure can also be applied in scenarios beyond audio mosaicing. One possibility is to examine whether supporting the development of diagonal structures in the activation matrix can also be beneficial when learning not only the activation matrix, but also the template matrix. Such an NMF procedure could be applied for learning and identifying repeating patterns in feature sequences, similar to [24] who used techniques based on NMF for this task. In this context, we hope that our approach may yield a simpler implementation as well as more flexibility since the maximal length of sequences does not need to be fixed.

Acknowledgments:

This work has been supported by the German Research Foundation (DFG MU 2686/6-1). The International Audio Laboratories Erlangen are a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer Institut für Integrierte Schaltungen. Furthermore, we would like to thank Colin Raffel and the other organizers of the *HAMR* Hack Day at ISMIR 2014, where the core ideas of the presented work were born.

6. REFERENCES

- [1] G. Bernardes. *Composing Music by Selection: Content-Based Algorithmic-Assisted Audio Composition*. PhD thesis, Faculty of Engineering, University of Porto, 2014.
- [2] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello. Medleydb: A multitrack dataset for annotation-intensive MIR research. In *Proc. of the 15th International Society for Music Information Retrieval Conference ISMIR*, pages 155–160, Taipei, Taiwan, October 2014.
- [3] G. Coleman, E. Maestre, and J. Bonada. Augmenting sound mosaicing with descriptor-driven transformation. In *Proc. of the International Conference on Digital Audio Effects (DAFx)*, Graz, Austria, 2010.
- [4] J. M. Comajuncos, A. Barrachina, J. O’Connell, and E. Gueas. Nuvolet: 3D gesture-driven collaborative audio mosaicing. In *Proc. of the International Conference on New Interfaces for Musical Expression*, pages 252–255, Oslo, Norway, 2011.
- [5] E. Costello, V. Lazzarini, and J. Timoney. A streaming audio mosaicing vocoder implementation. In *Proc. of the 16th International Conference on Digital Audio Effects (DAFx)*, Maynooth, Ireland, September 2013.
- [6] J. Driedger and M. Müller. TSM Toolbox: MATLAB implementations of time-scale modification algorithms. In *Proc. of the International Conference on Digital Audio Effects (DAFx)*, pages 249–256, Erlangen, Germany, 2014.
- [7] J. Driedger, T. Prätzlich, and M. Müller. Accompanying website: Let it bee – towards NMF-inspired audio mosaicing. <http://www.audiolabs-erlangen.de/resources/MIR/2015-ISMIR-LetItBee/>.
- [8] J. Eggert and E. Körner. Sparse coding and NMF. In *Proc. of the IEEE International Joint Conference on Neural Networks*, volume 4, pages 2529–2533, July 2004.
- [9] D. W. Griffin and J. S. Lim. Signal estimation from modified short-time Fourier transform. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 32(2):236–243, 1984.
- [10] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457–1469, 2004.
- [11] J. Janer and M. de Boer. Extending voice-driven synthesis to audio mosaicing. In *5th Sound and Music Computing Conference*, Berlin, Germany, July 2008.
- [12] J. Kim and H. Park. Toward faster nonnegative matrix factorization: A new algorithm and comparisons. In *Proc. of the IEEE International Conference on Data Mining (ICDM)*, pages 353–362, Pisa, Italy, 2008.
- [13] R. Kobayashi. Sound clustering synthesis using spectral data. In *Proc. of the International Computer Music Conference (ICMC)*, Singapore, 2003.
- [14] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Proc. of the Neural Information Processing Systems (NIPS)*, pages 556–562, Denver, USA, 2000.
- [15] M. Müller, V. Konz, W. Bogler, and V. Arifi-Müller. Saarland music data (SMD). In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR): Late Breaking session*, 2011.
- [16] J. Oswald. Plexure. CD, 1993. <http://www.allmusic.com/album/plexure-mw0000621108>.
- [17] N. Schnell, M. A. S. Cifuentes, and J.-P. Lambert. First steps in relaxed real-time typo-morphological audio analysis/synthesis. In *Sound and Music Computing*, Barcelona, Spain, 2010.
- [18] D. Schwarz. A system for data-driven concatenative sound synthesis. In *Proc. of the International Conference on Digital Audio Effects (DAFx)*, Verona, Italy, July 2000.
- [19] D. Schwarz. Concatenative sound synthesis: The early years. *Journal of New Music Research*, 35(1), March 2006.
- [20] P. Smaragdīs. Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In *Independent Component Analysis and Blind Signal Separation*, volume 3195 of *Lecture Notes in Computer Science*, pages 494–499. Springer Berlin Heidelberg, 2004.
- [21] P. Smaragdīs, B. Raj, and M. Shashanka. Sparse and shift-invariant feature extraction from non-negative data. In *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing ICASSP*, pages 2069–2072, Las Vegas, Nevada, USA, 2008.
- [22] P. A. Tremblay and D. Schwarz. Surfing the waves : Live audio mosaicing of an electric bass performance as a corpus browsing interface. In *Proc. of the International Conference on New Interfaces for Musical Expression*, pages 447–450, Sydney, Australia, September 2010.
- [23] T. Virtanen. Monaural sound source separation by non-negative matrix factorization with temporal continuity and sparseness criteria. *IEEE Transactions on Audio, Speech and Language Processing*, 15(3):1066–1074, 2007.
- [24] R. J. Weiss and J. P. Bello. Unsupervised discovery of temporal structure in music. *IEEE Journal of Selected Topics in Signal Processing*, 5:1240–1251, 2011.

REAL-TIME MUSIC TRACKING USING MULTIPLE PERFORMANCES AS A REFERENCE

Andreas Arzt, Gerhard Widmer

Department of Computational Perception, Johannes Kepler University, Linz, Austria
Austrian Research Institute for Artificial Intelligence (OFAI), Vienna, Austria

andreas.arzt@jku.at

ABSTRACT

In general, algorithms for real-time music tracking directly use a symbolic representation of the score, or a synthesised version thereof, as a reference for the on-line alignment process. In this paper we present an alternative approach. First, different performances of the piece in question are collected and aligned (off-line) to the symbolic score. Then, multiple instances of the on-line tracking algorithm (each using a different performance as a reference) are used to follow the live performance, and their output is combined to come up with the current position in the score. As the evaluation shows, this strategy improves both the robustness and the precision, especially on pieces that are generally hard to track (e.g. pieces with extreme, abrupt tempo changes, or orchestral pieces with a high degree of polyphony). Finally, we describe a real-world application, where this music tracking algorithm was used to follow a world-famous orchestra in a concert hall in order to show synchronised visual content (the sheet music, explanatory text and videos) to members of the audience.

1. INTRODUCTION

Real-time music tracking (or, score following) algorithms, which listen to a musical performance through a microphone and at any time report the current position in the musical score, originated in the 1980s (see [8, 24]) and still attract a lot of research [4, 6, 11, 15, 17, 21, 23]. In recent years this technology has already found use in real-world applications. Examples include Antescofo¹, which is actively used by professional musicians to synchronise a performance (mostly solo instruments or small ensembles) with computer realised elements, and Tonara², a music tracking application focusing on the amateur pianist and running on the iPad.

A common approach in music tracking, and also for the related task of off-line audio to score alignment (see

e.g. [9, 19, 20]), is to start from a symbolic score representation (e.g. in the form of MIDI or MusicXML). Often, this score representation is converted into a sound file using a software synthesizer. The result is a ‘machine-like’, low-quality rendition of the piece, in which we know the time of every event (e.g. note onsets). Then, a tracking algorithm is used to solve the problem of aligning the incoming live performance to this audio version of the score – thus, the problem of real-time music tracking can be treated as an on-line audio to audio alignment task.

In this paper we follow a similar approach, but instead of using the symbolic score directly, we propose to first automatically align a recording of another performance of the same piece to the score. Then, we use this automatically annotated ‘score performance’ as the new score representation for the on-line tracking process (for the related task of off-line performance to performance alignment see e.g. [18]). Our motivation for this is twofold. First of all, we expect the quality of the features to be higher than if they were computed from a synthesised version of the score. Also, in a performance a lot of intricacies are encoded that are missing in the symbolic score, including (local) tempo and loudness changes. In this way we implicitly also take care of special events like trills, which normally are insufficiently represented in a symbolic score representation.

As will be seen in this paper, this approach proves to be promising, but the results also depend heavily on which performance was chosen as a reference. To improve the robustness we further propose a multi-agent approach (inspired by [25]), where a related strategy was applied to off-line audio alignment), which does not depend on a single performance as a reference, but takes multiple ‘score performances’ and aligns the live performance to all these references simultaneously. The output of all agents is combined to come up with the current position in the score. As will be shown in the evaluation, this extension stabilises our approach and increases the alignment accuracy.

The paper is structured as follows. First, in Section 2 we give an overview on the data we use to evaluate our music tracker. For comparison, we then give results of the original tracking algorithm that our approach is based on in Section 3. In Section 4 we present a tracking strategy based on off-line aligned performances, which shows promising but unstable results. Then, in Section 5 we propose a multi-agent strategy, which stabilises the tracking process and



© Andreas Arzt, Gerhard Widmer.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Andreas Arzt, Gerhard Widmer. “Real-time Music Tracking using Multiple Performances as a Reference”, 16th International Society for Music Information Retrieval Conference, 2015.

ID	Composer	Piece Name	# Perf.	Groundtruth
CE	Chopin	Etude Op. 10 No. 3 (excerpt)	22	Match
CB	Chopin	Ballade Op. 38 No. 1 (excerpt)	22	Match
MS	Mozart	1 st Mov. of Sonatas KV279, KV280, KV281, KV282, KV283, KV284, KV330, KV331, KV332, KV333, KV457, KV475, KV533	1	Match
RP	Rachmaninoff	Prelude Op. 23 No. 5	3	Manual
B3	Beethoven	Symphony No. 3	1	Manual
M4	Mahler	Symphony No. 4	1	Manual

Table 1. The evaluation data set.

Error	CE	CB	MZ	RP	B3	M4
≤ 0.05	0.33	0.33	0.55	0.45	0.42	0.23
≤ 0.25	0.96	0.92	0.97	0.90	0.84	0.71
≤ 0.50	0.99	0.96	0.98	0.96	0.91	0.83
≤ 0.75	1	0.98	0.99	0.98	0.94	0.87
≤ 1.00	1	0.98	0.99	0.98	0.95	0.91

Table 2. Results for the *original on-line tracking algorithm*. The results are shown as proportion of correctly aligned pairs of time points (note times or downbeat times, respectively), for different error tolerances (in seconds). For instance, the first number in the first row means that for the Chopin Etude the alignment was performed for 33% of the notes with an error smaller than or equal to 0.05 seconds.

improves the results for all test pieces. Next, we compare the results of the previous chapters to each other (Section 6). Finally, we describe a real-life application of our algorithm at a world-famous concert hall, where it was used to track Richard Strauss' *Alpensinfonie* (see Section 7).

2. DATA DESCRIPTION

To evaluate a real-time music tracking algorithm, a collection of annotated performances is needed. Table 1 gives an overview on the data that will be used throughout the paper. It is important to note that the dataset includes two orchestral pieces (symphonies by Beethoven and Mahler), which in our experience are difficult challenges for music tracking algorithms, due to their high polyphony and complexity. The table also indicates how the ground truth was compiled. For the Chopin Ballade and Etude, and for the Mozart piano sonatas we have access to accurate data about every note onset ('matchfiles') that was played, as these were recorded on a computer-monitored grand piano (see [12] and [26] for more information about this data). For the Prelude by Rachmaninoff as well as for the Symphonies by Beethoven and Mahler we have to rely on manually annotated performances (at the note level for the prelude and at the downbeat level for the two symphonies).

Furthermore, we collected a number of additional performances of the pieces in our dataset. For these we do not have any annotations, and their sole purpose is to be

Error	CE	CB	MZ	RP	B3	M4
≤ 0.05	0.92	0.87	0.93	0.75	0.54	0.38
≤ 0.25	0.99	0.97	0.99	0.97	0.93	0.86
≤ 0.50	1	0.97	1	0.99	0.96	0.94
≤ 0.75	1	0.98	1	0.99	0.97	0.97
≤ 1.00	1	0.98	1	1	0.98	0.98

Table 3. Results for the *off-line alignments*. The results are shown as proportion of correctly aligned pairs of time points (note times or downbeat times, respectively), for different error tolerances (in seconds). For instance, the first number in the first row means that for the Chopin Etude the alignment was performed for 92% of the notes with an error smaller than or equal to 0.05 seconds.

processed fully automatically. These will act as replacements for the symbolic scores. We collected 7 additional performances for each piece in the dataset. We made an exception for the excerpts of the Ballade and the Etude by Chopin, as we already have 22 performances of those. We thus reused these performances accordingly, randomly selected 7 additional performances for each performance in the evaluation set, and treated them in the same way as the other additional data (i.e. we did not use any part of the ground truth, everything was computed automatically when they were used as a 'score performance'). We also took care not to use additional performances of the same performer(s) that occur in our evaluation set.

3. STANDARD MUSIC TRACKING BASED ON A SYMBOLIC SCORE REPRESENTATION

Our approach to music tracking is based on the standard dynamic time warping (DTW) algorithm. In [10] extensions to DTW were proposed that made it applicable for on-line music tracking: 1) the path is computed in an incremental way, and 2) the complexity is reduced to being linear in the length of the input sequences. Later on, this algorithm was extended with a 'backward-forward' strategy, which reconsiders past decisions, increasing the robustness [4], and a simple tempo model (see [3]), which greatly increases the ability of the algorithm to cope with tempo differences.

To make music tracking possible, some internal repre-

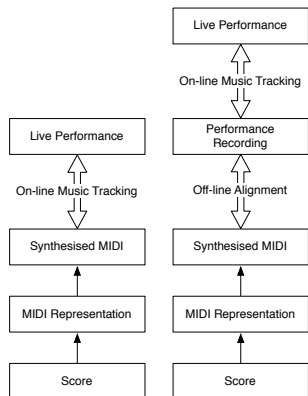


Figure 1. Standard music tracking (left) vs. music tracking via an off-line aligned reference performance (right).

sensation of the musical score is needed. In this case we start with a MIDI version of the score, which is converted into an audio file using a software synthesizer. Thus we actually treat this task as an audio-to-audio alignment problem, with additional knowledge about the score audio file (i.e. the exact timing of each note). See Figure 1 (left) for a sketch of this setup. In our approach we use the features (a mix of chroma features and ‘semi-tone onset’ features) and the distance computation method presented in [5].

For comparison, we re-evaluated this algorithm on our data. Each performance from our evaluation set was aligned to the symbolic score representation. The results are given in Table 2. The goal of this paper is to improve on these results, both regarding tracking precision and, especially, robustness (i.e. reduce the amount of big mistakes made by the music tracker). As can be seen, the algorithm works particularly well on the piano pieces, but shows problems with the two symphonies. A reason for this is that it is relatively easy to synthesise piano pieces from MIDI in acceptable quality, but it is much harder to do this automatically for orchestral pieces.

4. MUSIC TRACKING VIA A SINGLE PERFORMANCE AS A REFERENCE

As we are effectively treating the task of music tracking as an on-line audio-to-audio alignment task, we can actually use any annotated audio recording of a performance as a score representation. Using a real performance as a ‘score’ has some advantages.

First of all, an audio file synthesised from a deadpan MIDI file may sound bad compared to a real performance, thus also the features are of relatively low quality (i.e. they differ sometimes quite heavily from the features computed from the live performance we want to track). Despite obvious differences between performances, their respective

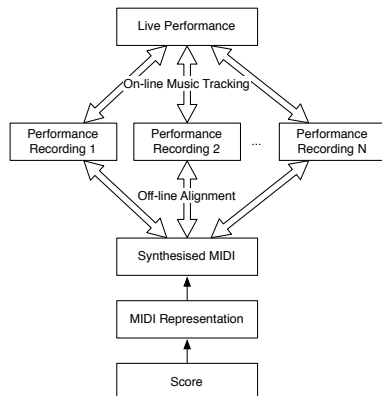


Figure 2. Multi-agent tracking based on off-line aligned performances as a reference.

features tend to be more similar to each other. This is especially true for orchestral pieces, which often include instruments that are hard to synthesise in high quality (or at least this would demand for expensive sound fonts and a lot of effort by a trained audio engineer).

Secondly, a performance implicitly encodes a lot of information that is missing in the symbolic score. This includes detailed information about tempo, loudness and articulation. Again we want to stress that of course performances differ from each other quite heavily, but compared to the differences between a performance and an audio synthesised from the MIDI, these differences are small.

There is also one big disadvantage: the symbolic information linking time points in the audio to beat times in the score, which we get for free when we use a MIDI file as the basis for the score audio, is missing. Thus, this information needs to be generated. There are two possible ways to do that: (1) by manual annotation, which can be very laborious, or (2) by automatic off-line alignment of the performance to the score – which is the option we decided on, as we are interested in an automatic method to improve tracking results (see Section 4.1 below).

Figure 1 shows a sketch of the intended setup. On the left, ‘normal’ music tracking is shown, where the live performance is aligned to the symbolic score (via a synthesised audio). On the right, another performance is first aligned to the symbolic score. This performance is then used as the new reference in the on-line alignment process.

4.1 Offline Alignment

To use a performance as a ‘score’ we have to generate the necessary symbolic information, linking time points in the audio to beat times in the score. As we are interested in an automatic way to improve the tracking results, we decided to use off-line audio alignment to align the ‘score perfor-

Error	CE	CB	MZ	RP	B3	M4
≤ 0.05	0.39	0.35	0.52	0.25	0.35	0.27
≤ 0.25	0.98	0.96	0.97	0.87	0.85	0.80
≤ 0.50	0.99	0.97	0.99	0.97	0.93	0.92
≤ 0.75	1	0.98	0.99	0.99	0.95	0.95
≤ 1.00	1	0.98	1	1	0.97	0.96

Table 4. Results for *on-line music tracking* based on a *single off-line aligned performance as a reference*. The results are shown as proportion of correctly aligned pairs of time points (note times or downbeat times, respectively), for different error tolerances (in seconds). For instance, the first number in the first row means that for the Chopin Etude the alignment was performed for 39% of the notes with an error smaller than or equal to 0.05 seconds.

mance’ to the symbolic score, which gives us the needed mapping as a result. As off-line audio alignment is far more accurate than on-line tracking, our intuition was that the increase in feature quality outweighs the introduced error by the off-line alignment process.

The off-line alignment is computed with the music tracking algorithm from Section 3 above, with the only difference being that in the end we compute the backward path, as it is done in the standard DTW algorithm. As this path is based on more information (i.e. it is computed in a non-causal way), the results are generally much more accurate than in the on-line case. Of course any off-line audio score alignment algorithm could be used for this task (see e.g. [16, 19, 20]).

Just to get a rough idea of how much error will be introduced by the off-line alignment, we ran an experiment on our test data and aligned it to the symbolic scores (later on, off-line alignments of the additional data will be used, but we expect a similar behaviour). Unsurprisingly, the results show that there is a gap between the results of the off-line approach (see Table 3) and the on-line music tracking approach (see Table 2). As we will use the off-line algorithm during data preparation, we strongly expect that the higher quality of the features and the additional information encoded in the performances will outweigh the error that is introduced during this step.

Thus, we aligned all the additional performances from Section 2 to the respective symbolic scores, resulting in performances with linked symbolic information. In the following sections, we will use these performances as new references (‘score performances’) for the music tracking algorithm.

4.2 Tracking based on an aligned Performance

Given the automatically computed ‘score performances’, we can now use them in the tracking process as shown in Figure 1. In this experiment, each performance from the evaluation set is aligned to the score via each respective ‘score performance’, resulting in 7 on-line alignments for each performance.

The results are given in Table 4 and should be compared

Error	CE	CB	MZ	RP	B3	M4
≤ 0.05	0.39	0.35	0.58	0.19	0.44	0.32
≤ 0.25	0.99	0.98	0.99	0.92	0.90	0.84
≤ 0.50	1	0.98	1	1	0.95	0.94
≤ 0.75	1	0.98	1	1	0.96	0.96
≤ 1.00	1	0.99	1	1	0.97	0.97

Table 5. Results for the *multi-agent tracking* approach based on a *set of off-line aligned performances as a reference*. The results are shown as proportion of correctly aligned pairs of time points (note times or downbeat times, respectively), for different error tolerances (in seconds). For instance, the first number in the first row means that for the Chopin Etude the alignment was performed for 39% of the notes with an error smaller than or equal to 0.05 seconds.

to the numbers in Table 2. As can be seen, the general trend is an improvement in robustness, especially for the complex orchestral pieces (e.g. the percentage of aligned downbeats with an error smaller than 250 ms increased from 71% to 80% for the Mahler Symphony).

Unfortunately, the results also proved to be unstable. Some performances are more similar (or at least easier to align) to each other, which also results in good tracking results – but the use of some of the ‘score performances’ led to results that were worse than our basic approach. A closer look at the positions where tracking errors occurred showed that some of them happened at the same points in time over all alignments of the piece – basically showing that some parts are harder to track than others. But there were also many alignment errors that occurred only for one or two of the ‘score performances’, but not for the others. This led us to the idea to combine individual on-line alignments in such a way, that it would smooth out these errors.

5. MUSIC TRACKING VIA A SET OF PERFORMANCES AS REFERENCE

The analysis of the results from Section 4 above showed that a combination of a number of on-line alignments might further improve the tracking results. Here, we propose a simple multi-agent strategy (see Figure 2 for an illustration). During a live concert n trackers run in parallel and each tracker tries to align the incoming live performance to its score representation, each producing its own, independent hypothesis of the current position in the score. Finally, the hypotheses are combined to form one collective hypothesis of the music tracking system.

Many different ways of combining the hypotheses would be possible, e.g. based on voting or on the current alignment error of the individual trackers. Here, we decided on a very simple method: taking the median of the positions that are returned by the individual trackers. The reasoning behind this is that trackers tend to make mistakes in both directions – i.e. ‘running ahead’ (reporting events to early), and ‘lagging behind’ (reporting events with some delay) –

with about the same frequency. Thus, trackers that stay safely in the middle of the pack tend to give a robust estimate of the position in the score.

Furthermore, using the median also means that as long as $\frac{n}{2} + 1$ trackers stay close to the actual position, the system would still come up with a reasonable position estimate – while this is not directly reflected in the evaluation results, this extra robustness is convenient when the tracking algorithm is used in real-world applications. Further strategies to increase the robustness are possible, like the automatic replacement of trackers that got lost, but were not used in our experiments.

For the evaluation we set $n = 7$, as this was a good trade-off between robustness and computation time (7 on-line alignments can still be easily computed in real-time on a conventional consumer laptop). The results, given in Table 5, show that our approach is working well. Errors of more than 1 second are rare, and the multi-agent approach even improved the alignment precision for all pieces (with the exception of the Prelude by Rachmaninoff).

6. DISCUSSION

The main goal of our approach was to increase the robustness of the algorithm, i.e. to decrease the frequency of ‘large’ errors and to make sure that the tracker does not get lost, even when following difficult orchestral pieces. For convenience, we give a summary of the results (see Table 6) based on a common measure in the evaluation of music tracking algorithms: the percentage of notes that were aligned with an error less than or equal to 250 ms (see [7]). As can be seen, the multi-agent approach based on automatically aligned reference performances improves the results heavily – in fact for CB the results of the on-line alignment even surpassed the off-line alignment. For the results on the Chopin data (CE and CB) one has to take into account that we used 22 performances which were recorded by different performers, but still on the same piano and with the same recording setup, which will have a positive influence on the alignment results. Still, as the remaining results show, even when completely unrelated performances of the same piece were used as references, the alignment results improved drastically.

Especially for the orchestral pieces (B3 and M4), we can see that our intuition proved to be correct: the error introduced by the off-line alignment had a lot less impact than the better quality of the features and the additional tempo and loudness information provided by the performances. In addition, the multi-agent approach proved to be very effective regarding the increase in robustness. It smooths out some of the bigger errors that occur when using just a single performance as a score reference.

7. REAL-LIFE SCENARIO: MUSIC TRACKING IN THE CONCERTGEBOUW AMSTERDAM

The multi-national European research project PHENICX³ provided us with the unique opportunity (and challenge) to

Piece	Offline	Standard	Via 1	Via 7
CE	99.06%	95.62%	97.92%	98.78%
CB	97.13%	92.10%	96.00%	97.93%
MZ	99.35%	96.88%	97.46%	99.04%
RP	96.62%	90.14%	87.47%	92.47%
B3	92.88%	83.67%	85.04%	89.55%
M4	86.74%	71.15%	80.06%	83.66%

Table 6. Comparison of the results (error tolerance 250 ms). The results are shown as percentage of matching pairs of time points (note times or downbeat times, respectively). For instance, the first number in the first row means that for the Chopin Etude the off-line alignment was performed for 99.06% of the notes with an error smaller than or equal to 0.25 seconds. The results of the *offline* alignment algorithm are only shown for comparison. *Standard* refers to the basic on-line music tracker (see Section 3), *Via 1* to the tracker using a single ‘score performance’ as a reference, *Via 7* to the multi-agent approach based on 7 trackers.

demonstrate our score following technology in the context of a big, real-life symphonic concert (for a full description of this experiment see [2], a similar study was presented in [22]). The general goal of the project is to develop technologies that enrich the experience of classical music concerts. In the experiment to be described, this was done by using the live performance tracker to control, in real time and via WiFi, the transmission and display of additional visual and textual information, synchronised to the live performance on stage. The user interface and the visualisations were provided by our project partner Videodock⁴. Some impressions can be seen in Figure 3.

The event took place on February 7th, 2015, in the Concertgebouw in Amsterdam. The Royal Concertgebouw Orchestra, conducted by Semyon Bychkov, performed the *Alpensinfonie* (Alpine Symphony) by Richard Strauss. This concert was part of a series called ‘Essentials’, during which technology developed within the project can be tested in a real-life concert environment. All the tests during this concert series have to be as non-invasive as possible. For the demonstration during the concert in question, a test audience of about 30 people was provided with tablet computers and placed in the rear part of the concert hall.

In contrast to the experiments presented in this paper so far, we did not even have access to a symbolic score. Instead, we annotated a single performance manually (on the level of downbeats) and used it as a score representation. Then, to add extra robustness, we aligned 6 more performances to this reference, resulting in 7 instances that can be used for the tracking process.

The event in the Concertgebouw was a big success. The tracking went smoothly and there were no glitches, only some minor inaccuracies, and the accuracy was more than sufficient to trigger the visualisation in time.

After the event we annotated an audio recording of the concert to be able to perform quantitative experiments (see

³<http://phenicx.upf.edu>

⁴<http://videodock.com>

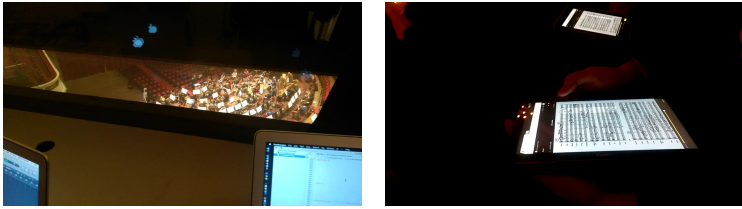


Figure 3. Left: View from the control room onto the stage (during orchestra rehearsal); right: synchronised score display in the audience during the concert.

Err. (sec)	Single	Multi-agent
≤ 0.25	78.25%	81.80%
≤ 0.50	92.20%	93.24%
≤ 0.75	95.57%	96.44%
≤ 1.00	97.49%	98.01%

Table 7. Real-time alignment results for the single tracker (using only on manually annotated performance), and the multi-agent tracker, shown as percentages of correctly aligned pairs of downbeats. For instance, the first number in the first row means that the single tracker aligned 78.25% of the downbeats with an error smaller than or equal to 0.25 seconds.

Table 7). The first column shows the results of the tracking using only the manually annotated performance as a reference. The second column shows the results of the multi-agent approach. Also in this case using multiple performances as a reference improved the tracking results: extra robustness and a slight increase in accuracy were achieved without any extra manual efforts as the additional data was prepared by automatic methods.

8. CONCLUSION

In this paper we presented an alternative approach to real-time music tracking. Instead of tracking directly on a symbolic score representation, we first use off-line alignment to match other performances of the piece in question to the symbolic score. We then use these performances as our new score representation, which results in high quality features, and implicitly also adds extra information about how this piece generally is performed. Together with a multi-agent tracking strategy, which smooths out most of the major errors, we achieve increased robustness and also increase the accuracy of the live tracking, especially for complex orchestral music. We also reported on a successful real-world test of our algorithm in a world-famous concert hall.

In the future, we will also look at other options to combine tracking results of the individual trackers. While taking the median seems like a natural choice, more sophisticated strategies also based on alignment costs might be

promising. A further problem which deserves a closer look is the automatic selection strategy of the ‘score performances’. For this paper we simply decided on 7 additional performances of the pieces based on availability. With a bigger database, automatic selection of the ‘best score performances’ for an on-going live performance becomes an interesting question, and a good selection strategy might further improve the tracking results.

A common problem of real-time music tracking and audio to score alignment are structural differences between the score and the performance. For example, if a piece has some repeated sections, the performers might decide to play the repetition or to leave it out. For the experiments in this data we chose the additional ‘score performances’ manually, such that they have the same structure as the piece we want to track, but in the future we will try to cope with this automatically – in the preparation phase via the technique used in [13] or [14] (maybe in combination with the method described in [25], to bring the benefit of using multiple performances also to the preprocessing stage), and in the live tracking phase with the approach presented in [1], extended to orchestral music.

9. ACKNOWLEDGEMENTS

This research is supported by the Austrian Science Fund (FWF) under project number Z159 and the EU FP7 Project PHENICX (grant no. 601166).

10. REFERENCES

- [1] Andreas Arzt, Sebastian Böck, Sebastian Flossmann, Harald Frostel, Martin Gasser, and Gerhard Widmer. The complete classical music companion v0.9. In *Proc. of the AES Conference on Semantic Audio*, London, England, 2014.
- [2] Andreas Arzt, Harald Frostel, Thassilo Gadermaier, Martin Gasser, Maarten Grachten, and Gerhard Widmer. Artificial intelligence in the concertgebouw. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, Buenos Aires, Argentina, 2015.
- [3] Andreas Arzt and Gerhard Widmer. Simple tempo models for real-time music tracking. In *Proc. of*

- the Sound and Music Computing Conference (SMC)*, Barcelona, Spain, 2010.
- [4] Andreas Arzt, Gerhard Widmer, and Simon Dixon. Automatic page turning for musicians via real-time machine listening. In *Proc. of the European Conference on Artificial Intelligence (ECAI)*, Patras, Greece, 2008.
- [5] Andreas Arzt, Gerhard Widmer, and Simon Dixon. Adaptive distance normalization for real-time music tracking. In *Proc. of the European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania, 2012.
- [6] Arshia Cont. A coupled duration-focused architecture for realtime music to score alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(6):837–846, 2009.
- [7] Arshia Cont, Diemo Schwarz, Norbert Schnell, and Christopher Raphael. Evaluation of real-time audio-to-score alignment. In *Proceedings of the International Conference on Music Information Retrieval (ISMIR)*, Vienna, Austria, 2007.
- [8] Roger Dannenberg. An on-line algorithm for real-time accompaniment. In *Proc. of the International Computer Music Conference (ICMC)*, Paris, France, 1984.
- [9] Roger Dannenberg and Ning Hu. Polyphonic audio matching for score following and intelligent audio editors. In *Proc. of the International Computer Music Conference (ICMC)*, Singapore, 2003.
- [10] Simon Dixon. An on-line time warping algorithm for tracking musical performances. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, Edinburgh, Scotland, 2005.
- [11] Zhiyao Duan and Bryan Pardo. A state space model for on-line polyphonic audio-score alignment. In *Proc. of the IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011.
- [12] Sebastian Flossmann, Werner Goebel, Maarten Grachten, Bernhard Niedermayer, and Gerhard Widmer. The magaloff project: An interim report. *Journal of New Music Research*, 39(4):363–377, 2010.
- [13] Christian Fremery, Meinard Müller, and Michael Clausen. Handling repeats and jumps in score-performance synchronization. In *Proc. of the International Conference on Music Information Retrieval (ISMIR)*, Utrecht, The Netherlands, 2010.
- [14] Maarten Grachten, Martin Gasser, Andreas Arzt, and Gerhard Widmer. Automatic alignment of music performances with structural differences. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, Curitiba, Brazil, 2013.
- [15] Filip Korzeniowski, Florian Krebs, Andreas Arzt, and Gerhard Widmer. Tracking rests and tempo changes: Improved score following with particle filters. In *Proc. of the International Computer Music Conference (ICMC)*, Perth, Australia, 2013.
- [16] Marius Miron, Julio José Carabias-Orti, and Jordi Janer. Audio-to-score alignment at note level for orchestral recordings. In *Proc. of the International Conference on Music Information Retrieval (ISMIR)*, Taipei, Taiwan, 2014.
- [17] Nicola Montecchio and Arshia Cont. A unified approach to real time audio-to-score and audio-to-audio alignment using sequential monte-carlo inference techniques. In *Proc. of the IEEE Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Prague, Czech Republic, 2011.
- [18] Meinard Müller, Frank Kurth, and Michael Clausen. Audio matching via chroma-based statistical features. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, London, Great Britain, 2005.
- [19] Meinard Müller, Frank Kurth, and Tido Röder. Towards an efficient algorithm for automatic score-to-audio synchronization. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, Barcelona, Spain, 2004.
- [20] Bernhard Niedermayer and Gerhard Widmer. A multi-pass algorithm for accurate audio-to-score alignment. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, The Netherlands, 2010.
- [21] Takuma Otsuka, Kazuhiro Nakadai, Toru Takahashi, Tetsuya Ogata, and Hiroshi G. Okuno. Real-time audio-to-score alignment using particle filter for coplayer music robots. *EURASIP Journal on Advances in Signal Processing*, 2011(2011:384651), 2011.
- [22] Matthew Prockup, David Grunberg, Alex Hrybyk, and Youngmoo E. Kim. Orchestral performance companion: Using real-time audio to score alignment. *IEEE Multimedia*, 20(2):52–60, 2013.
- [23] Christopher Raphael. Music Plus One and machine learning. In *Proc. of the International Conference on Machine Learning (ICML)*, Haifa, Israel, 2010.
- [24] Barry Vercoe. The synthetic performer in the context of live performance. In *Proc. of the International Computer Music Conference (ICMC)*, Paris, France, 1984.
- [25] Siying Wang, Sebastian Ewert, and Simon Dixon. Robust joint alignment of multiple versions of a piece of music. In *Proc. of the International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014.
- [26] Gerhard Widmer. Discovering simple rules in complex data: A meta-learning algorithm and some surprising musical discoveries. *Artificial Intelligence*, 146(2):129–148, 2003.

TWO DATA SETS FOR TEMPO ESTIMATION AND KEY DETECTION IN ELECTRONIC DANCE MUSIC ANNOTATED FROM USER CORRECTIONS

Peter Knees,¹ Ángel Faraldo,² Perfecto Herrera,² Richard Vogl,¹
Sebastian Böck,¹ Florian Hörschläger,¹ Mickael Le Goff³

¹ Department of Computational Perception, Johannes Kepler University, Linz, Austria

² Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain

³ Native Instruments GmbH, Berlin, Germany

peter.knees@jku.at

ABSTRACT

We present two new data sets for automatic evaluation of tempo estimation and key detection algorithms. In contrast to existing collections, both released data sets focus on electronic dance music (EDM). The data sets have been automatically created from user feedback and annotations extracted from web sources. More precisely, we utilize user corrections submitted to an online forum to report wrong tempo and key annotations on the *Beatport* website. *Beatport* is a digital record store targeted at DJs and focusing on EDM genres. For all annotated tracks in the data sets, samples of at least one-minute-length can be freely downloaded. For key detection, further ground truth is extracted from expert annotations manually assigned to *Beatport* tracks for benchmarking purposes. The set for tempo estimation comprises 664 tracks and the set for key detection 604 tracks. We detail the creation process of both data sets and perform extensive benchmarks using state-of-the-art algorithms from both academic research and commercial products.

1. INTRODUCTION

Electronic dance music (EDM) is one of the most important and influential music genres of our time. The genre has been defined as a broad category of popular music that, since the end of the 1990s, encompasses styles such as techno, house, trance, and dubstep, and, uniquely, utilizes electronic instruments such as synthesizers, drum machines, sequencers, and samplers. Traditionally, technologically-mediated live performances form an integral part of EDM [6, 8].

Historically, EDM evolved from and links genres from the 1950s to the 1980s such as soul, funk, disco, rap, and techno. After two decades of isolation as a genre, today, we are witnessing how it not only influences its legitimate

forerunner genres, but also most generic and formulaic pop forms, including contemporary rock, r&b and rap music. In fact, given its spread over millions of followers, EDM is a central element in the 21st century's popular music — and therefore a major economical factor in the entertainment industry. ^{1 2 3}

Despite its popularity, in terms of musical sophistication, the reputation of EDM might not be the best: “simplistic,” “too repetitive,” “feasible with lack of talent,” “fake music,” or “button-pushing” are some of the criticisms we can find in press, social media, or even in academia. In contrast to such stereotyped views, for MIR research, EDM, in fact, presents an interesting area as some styles have inherent properties that may challenge or pose difficult problems for existing music description algorithms. These properties include complex rhythm patterns (as can be observed in IDM or breakbeat), tonal patterns beyond major-minor distinctions [41], structural development not using intro-verse-chorus, temporal developments simply based on reoccurring tension-relaxation patterns (such as “drops” [1, 43]), or, contrarily, developments that are not built on tension-relaxation schemes at all. This has been acknowledged by musicologists and theorists [8, 19, 40, 41, 44].

Although some work on topics pertinent to electronic music, e.g., regarding timbre, rhythm, segmentation, or individual sub genres, has been published in recent years [1, 10, 12, 17, 18, 26, 29–31, 33, 35, 42, 43], and there seems to be a trend towards tempo estimation, e.g., [20, 28], we still lack EDM-specific annotated collections and data sets. For instance, existing data sets for tempo (or beat) estimation comprise of ballroom dance genres [23], Beatles tracks [13, 25], classical, jazz, and (J-)pop [22], rock/pop, dance, classical, folk and jazz [24], or examples from classical music, romantic music, film soundtracks, blues, chanson, and solo guitar tracks selected for “difficulty” [27]. Similarly, for tonality-related tasks, existing data sets comprise of tracks by The Beatles and Queen [32], Robbie Williams [16], piano chords [2], and rock and pop mu-

 © Peter Knees, Ángel Faraldo, Perfecto Herrera, Richard Vogl, Sebastian Böck, Florian Hörschläger, Mickael Le Goff.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Peter Knees, Ángel Faraldo, Perfecto Herrera, Richard Vogl, Sebastian Böck, Florian Hörschläger, Mickael Le Goff. “Two data sets for tempo estimation and key detection in electronic dance music annotated from user corrections”, 16th International Society for Music Information Retrieval Conference, 2015.

¹ http://www.amsterdam-dance-event.nl/static/files/dance-economics_economic-significance-edm-17102012.pdf

² <http://www.thembj.org/2013/12/the-economics-of-the-electronic-dance-industry/>

³ <https://smartasset.com/insights/the-economics-of-electronic-dance-music-festivals>

sic [7, 15]. Other data sets used in MIR research that contain electronic dance music or other types of electronic music, such as the Million Song Dataset [3], the MediaEval 2014 Crowdsourcing Task data set,⁴ or the art-oriented UbuWeb corpus [11], lack human annotations of tempo and key, among others.

In this paper, we want to address this lack of EDM data sets for MIR research. To this end, we propose two data sets – one for the task of tempo estimation and one for the task of key detection. In contrast to existing collections, both released data sets focus on electronic dance music. Since labeling a corpus manually is a labor-intensive task, we follow another strategy to obtain human ground truth annotations for tracks from a digital online record store focusing on EDM, namely *Beatport*.⁵ As tempo and key information given by the retailer are imperfect, users were encouraged to give feedback on spotted incorrect data using a dedicated online forum. We describe this forum in Section 2. We extract the contained information using regular expressions and knowledge-based filtering in order to obtain user-based annotations for the corresponding tracks (Section 3). In Section 4, we present some descriptive statistics on the extracted ground truth. Section 5 reports on benchmarking results obtained using a variety of academic and commercial algorithms on the two new data sets. We conclude this paper by discussing the modalities of making this data set available to the research community and by drawing conclusions in Section 6.

2. BEATPORT USER FORUM

Beatport is a US- and Germany-based online music store targeted at DJs and music producers. In comparison to standard music web stores, it emphasizes additional meta-data relevant for DJs, such as tempo, key, and style, as well as information on record label, release information, version, and remixing artists, making it an interesting source for MIR research. Meta-data associated with a track can be easily extracted in JSON format from the source code of the corresponding web page. This meta-data also contains links to the listening snippets of the tracks, which are typically between 60 and 120 seconds long.

An important observation is that tempo and key information provided on the website are determined algorithmically upon upload of the tracks by undisclosed algorithms. Thus, this information can not be considered a ground truth and is therefore useless for evaluation purposes.⁶ However, apparently being aware of the imperfection of their automatic annotation algorithms, until late 2014, *Beatport* asked its customers to provide feedback on tempo and key information via a link (“Report Incorrect BPM/Key”) pointing to a dedicated online forum. In this forum, users would post their corrections in free-form text using natural language, i.e., the feedback given is highly

“93 bpm not 111 or whatever it is!”
 “bpm is 120 not 160. i should know, i made it :)”
 “173 bpm / g minor”
 “key should be c# minor”
 “wrong bpm”
 “the bpm is fine... its the genre. it’s progressive house, not tech house.”

Table 1. Examples of correctional comments published on the online forum (links to tracks removed for readability)

heterogeneous and in many cases incomplete (no information, reference to track missing, etc.)⁷ Nonetheless, as other work has shown [37], online forums present a great opportunity to extract user-generated, music-related information. Table 1 shows typical comments posted into the forum.

We performed a complete web crawl of this user forum in May 2014. At the time of the crawl, there were 2,412 comments available, of which 1,857 contained a direct link to a track on the *Beatport* website. From the link to the track, we download the complete meta-data record in JSON format using web scraping techniques. From this, we also extract the associated style descriptor for statistical reasons, cf. Section 4.

3. GROUND TRUTH EXTRACTION

In this section we detail the process of extracting ground truth from the 1,857 comments that contained a link to a track. First, we describe the process of extracting BPM (beats-per-minute) information. Second, we describe the extraction of key information from the forum, as well as from expert sources available online. All steps were performed after case-folding the texts.

3.1 BPM Extraction

For BPM extraction, we retain all posts that contain the word ‘bpm’ and a two- or three-digit number, optionally followed by a decimal point and a one- to three-digit number. On the remaining posts, we apply several rule-based filter criteria to exclude unlikely or possibly unrelated numbers. This comprises of all numbers below 40 and above 250 as these represent tempo values with a low probability of occurrence in this context. Furthermore, we remove all two- or three-digit numbers (with optional decimal places) that are preceded by the word ‘not’ as well as the number representing the tempo given by the *Beatport* website (as this is obviously the wrong tempo). We then take the first matching number as ground truth for the linked track. Applying this restrictive filtering, we were able to extract 726 records of BPM tempo annotations that were made by humans rather than an algorithm.

⁴ <https://osf.io/h92g8/>

⁵ <http://www.beatport.com>

⁶ The same holds for the associated genre/style information, which has to be set by the human uploading the tracks onto the platform and often results in rather arbitrary assignments, cf. [38, 39]

⁷ The resulting difficulty in exploiting this information might be one of the reasons why none of the reported errors have led to a correction of the meta-data on the *Beatport* website, which has been also negatively commented on by users, and could be a reason for discontinuing this form of feedback.

From these 726 annotations, we identified duplicate BPM entries for the same IDs (e.g., when different users report a wrong tempo on the website for the same track or when the same user repeatedly urges to incorporate a suggestion made before). Furthermore, we use audio fingerprinting as well as manual inspection in order to map duplicate audio files with different IDs to one single ID (9 files). This joint information on duplicates is used to substantiate the tempo annotation: if there is more than one tempo correction available per track, we put all candidates with the same tempo (within $\pm 4\%$) into a bin and consider the mean of all tempo candidates within the bin that contains the absolute majority ($> 50\%$) as the correct annotation. If no such bin exists, the track is rejected. Since this was only the case for one file, we manually set the correct tempo for this file. This way 61 entries, including the 9 files with same audio but different IDs, were removed. In total, 42 resulting ground truth annotations are based on multiple sources. We further removed one file because the linked mp3 sample was no longer available. After this procedure, we obtain a human-annotated data set of 664 distinct electronic music tracks.

3.2 Key Extraction

A similar process was carried out on the same data, in order to extract user corrections on *Beatport*'s key tags. Additionally, we found three independently annotated sources that use the *Beatport* database for software benchmarking.

3.2.1 User Forum

In the 1,857 posts that contained a link to a track, we filter all that contain the sequences 'mixed-in-key,' 'mixed in key,' 'mik,' and 'melodyne' in order to exclude posts reporting on other algorithms' outputs. In the remaining posts, we search for occurrences of the regular expression `[a-g](\s*(#|b|sharp|flat))?\s*(min|max)(or)?` where `\s` represents the class of whitespace characters. Additionally, all occurrences of this expression preceded by the word 'not' are excluded as well as matches that represent the same key as the key indicated in the *Beatport* meta-data (which, again, is obviously wrong).

After processing, we found a total of 404 key corrections which can be regarded as ground truth. In this group we found 15 duplicates and one track which is no longer available, leaving us with a total of 388 global-key annotations.

3.2.2 DJ Endo Labels

In order to compare different commercial key detection approaches, *DJ Endo* has published two online reports with different samples from *Beatport* that are built on his own ground truth annotations. For the first report (2011),⁸ he annotates 100 songs, providing a (slightly truncated) GIF image file of the list. This image contains 99 items (one of which is a duplicate) with artist name, song title, his personal annotation, and the predictions of *Mixed-In-Key* and *Beatport*. We used OCR software to convert this list to

⁸ <http://blog.dubspot.com/dubspot-lab-report-mixed-in-key-vs-beatport>

a spreadsheet in order to obtain the human labels and access to the audio excerpts from the *Beatport* website. Using a simple script that queries the *Beatport* search page for artist and title, we retrieve the meta-data of candidate tracks. In case artist and title match perfectly, they are assigned, in case there are multiple candidates (e.g., different remix versions), a manual assignment to the correct version is done. Ultimately, this allowed us to obtain 92 out of the unique 98 tracks in the list image.

In the second report (2013),⁹ DJ Endo makes a more exhaustive comparison between 7 different key estimation applications, including the *Beatport* database. The track list holds a total of 119 songs, 19 of which come from *YouTube* videos, while 7 tracks are listed without any link or *Beatport* key tag. We have excluded these 26 items, obtaining a batch of 93 songs with ground truth and links to the *Beatport* samples.

3.2.3 DJTechTools Labels

A third internet source (2014)¹⁰ provides ground truth from human consensus for another 60 tracks. Besides the manual annotations and the *Beatport* key tags and links, 10 commercial products are evaluated.

Two of the annotations in this collection provide two key estimates per track. These have been checked and reduced to a single key manually by one of the authors, to fit with the rest of the collection.

3.2.4 Unification

With all these sources added together, we obtain a compound data set with 633 annotated tracks. However, we found a total of 29 duplicates among the different sources. In all cases, the different sources agree on the reported key, giving evidence that our approach is working (see also Section 6). This leaves us with a global-key detection data set of 604 EDM excerpts.

4. DATA SET CHARACTERISTICS

In this section we want to analyze the newly obtained data sets. To this end, we present descriptive statistics and also utilize the style information extracted from the *Beatport* meta-data. Please note that this style information does not represent a consistently annotated ground truth but merely serves as a broad reference to estimate the characteristics of the data sets.

4.1 Tempo Data Set Statistics

The Tempo data set contains tempo ground truth for 664 samples. Table 2 provides descriptive statistics for the samples within the different *Beatport* styles. The table contains the corresponding number of samples as well as the minimum, the maximum, the mean (\bar{x}), the median (\tilde{x}) and the standard deviation (σ) of the tempo annotations for each individual style. The extracted tempo ground truth

⁹ <http://blog.dubspot.com/endo-harmonic-mixing-key-detection-analysis>

¹⁰ <http://www.djtechtools.com/2014/01/14/key-detection-software-comparison-2014-edition>

style	#	\bar{x}	\tilde{x}	σ	min	max
reggae-dub	2	70.0	70.0	0.0	70.0	70.0
chill-out	15	88.3	80.0	27.0	53.0	173.0
indie-dance-nu-dsc.	11	97.9	99.0	16.6	80.0	123.0
hip-hop	2	107.5	107.5	32.5	75.0	140.0
glitch-hop	17	109.9	110.0	26.9	80.0	174.0
deep-house	24	120.1	122.0	8.3	82.0	126.0
house	23	120.3	126.0	26.9	58.0	174.0
tech-house	22	123.8	126.0	5.3	107.0	130.0
techno	61	126.1	126.0	13.7	63.5	180.0
minimal	8	126.8	127.5	1.6	123.0	128.0
progressive-house	19	126.8	128.0	8.4	96.0	140.0
electronica	54	127.2	129.0	32.7	64.0	180.0
dj-tools	9	128.0	126.0	21.0	93.0	175.0
electro-house	22	129.4	128.0	21.7	63.0	175.0
funk-r-and-b	1	135.0	135.0	0.0	135.0	135.0
hard-dance	8	135.1	148.0	27.2	90.0	171.4
dubstep	76	135.2	140.0	23.7	70.0	180.0
breaks	26	138.9	140.0	14.4	83.5	170.0
trance	74	140.3	140.0	7.3	130.0	199.0
psy-trance	34	143.6	146.5	17.2	85.0	190.0
pop-rock	3	144.0	130.0	21.2	128.0	174.0
drum-and-bass	139	162.0	173.0	28.0	80.0	180.0
hardcore-hard-tech.	14	174.6	171.2	14.7	140.0	200.0
all	664	136.7	140.0	28.3	53.0	200.0

Table 2. Statistics for the *GiantSteps* Tempo data set per style (#...number of examples, \bar{x} ...mean BPM value, \tilde{x} ...median, σ ...std.dev.).

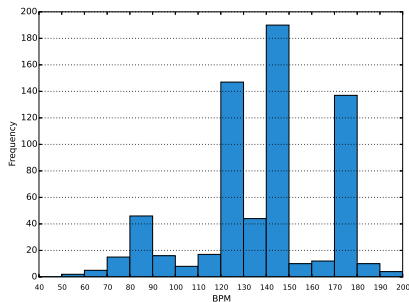


Figure 1. Distribution of BPM values in the Tempo set.

ranges from 53 to 200 BPM. Figure 1 contains a histogram of all BPM values in the data set. It reveals that most of the values are between 120 and 150 BPM, furthermore a peak between 170 and 180 BPM is apparent. This peak can most likely be attributed to the style *drum-and-bass* ($\bar{x} = 162$ BPM, $\tilde{x} = 173$ BPM), which makes up 20.9% of all samples. This style is known for very high tempos (above 160 BPM) and seems to be a challenging and error prone task for beat and tempo estimation algorithms due to its syn-copated beat structure. The evaluation of different tempo estimation approaches presented in Section 5 supports this theory. We argue that *Beatport*'s algorithmic issues with this genre, apart from the style's popularity, are the reason that many incorrect estimates were found by users and reported. Figure 2 visualizes the distribution of the different *Beatport* styles in the data set by means of a histogram.

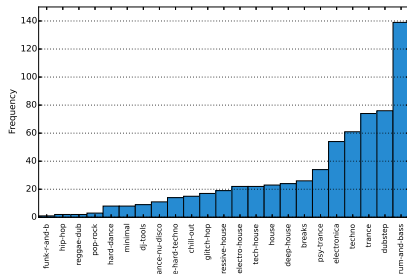


Figure 2. Histogram of tracks per style in the Tempo set.

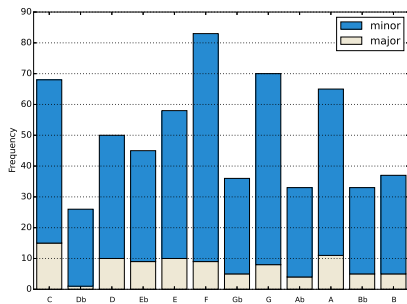


Figure 3. Distribution of keys in the Key data set.

4.2 Key Data Set Statistics

The Key data set contains 604 tracks with ground truth. Table 3 presents some simple statistics, including number of excerpts per subgenre, percentage of major and minor keys and most frequent key. 84.8% of the data set is in minor. Figure 3 shows the distribution of the corpus by tonal centers. The most frequent key is *Fm*, closely followed by *Cm* and *Gm*. Overall the distribution of tonics is relatively balanced. This is possibly related to the modes of production of these styles of music.

Figure 4 presents a histogram of tracks arranged by *Beatport* genre tags. We observe that these are unevenly distributed, with 344 excerpts (57%) pertaining to different "house" styles, whereas other subgenres and categories are underrepresented, with only 3 to 6 tracks each (funk and r&b, glitch-hop, hard-dance, hardcore, hip-hop, psy-trance, reggae/dub, and dj-tools).

5. BENCHMARKING

In this section we provide benchmarking results for both academic and commercial approaches on both data sets to estimate the performance of current methods as well as getting an impression of the "difficulty" of the data sets.

style	#	maj (%)	min (%)	most freq. key (%)
breaks	14	28.6	71.4	C (21.0)
chill-out	11	36.3	63.6	Em, Dm, Ab (18.1)
deep-house	77	5.2	94.8	Cm (13.0)
dj-tools	3	33.3	66.7	—
drum-and-bass	38	18.4	81.6	Gm (28.9)
dubstep	22	9.1	90.9	Fm (22.7)
electro-house	51	9.8	90.2	Fm (25.5)
electronica	20	20.0	80.0	Fm (20.0)
funk-r-and-b	3	0.0	100.0	—
glitch-hop	6	20.0	80.0	Gm (15.0)
hard-dance	4	0.0	100.0	Gbm (50.0)
hardcore-hard-tech.	3	33.3	66.7	—
hip-hop	4	0.0	100.0	Em (50.0)
house	47	17.0	83.0	Gm,Cm (12.8)
indie-dance-nu-dsc.	14	21.4	78.6	—
minimal	11	0.0	100.0	Em,Am (27.3)
pop-rock	7	57.1	42.9	Gm (42.9)
progressive-house	88	21	67	Am (12)
psy-trance	5	0.0	100.0	Fm (40.0)
reggae-dub	3	33.3	66.7	—
tech-house	81	12.4	87.6	Dm (14.9)
techno	34	17.6	82.4	Cm (17.6)
trance	58	12.0	88.0	Fm (24.1)
all	604	15.2	84.8	Fm (12.0)

Table 3. Statistics for the *GiantSteps* Key data set per style (number of examples, percentage of major and minor keys, most frequent key)

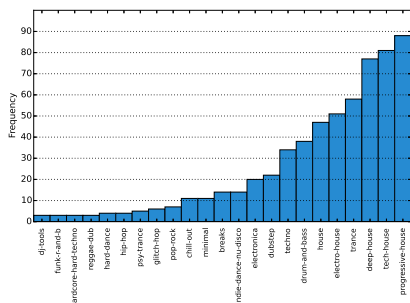


Figure 4. Histogram of tracks per style within the *GiantSteps* Key data set.

5.1 *GiantSteps* Tempo Data Set

Common in tempo estimation tasks, results are provided as accuracies within a $\pm 4\%$ tolerance window. *Accuracy1* considers an estimate to be correct if it is within $\pm 4\%$ of the true tempo. *Accuracy2* also considers an estimate to be correct if it is within $\pm 4\%$ of either a third, half, double or triple of the true tempo, thus being permissive of so-called octave errors.

5.1.1 Algorithms

As a *baseline* we evaluate the annotations created by *Beatport*'s undisclosed algorithm, obtained by evaluating the tempo annotations that were initially reported as incorrect. We expect very low values from this strategy, as the data set consists only of cases where *Beatport* has given wrong

estimates. However, the results are not trivially all zero, as the tolerance window allows for correct results if only minor deviations were corrected, as well as for corrections of octave errors.

In terms of academic algorithms we evaluate the tempo estimation approaches by Davies and Plumbly [14], Böck et al. [4], Gkiokas et al. [21], Percival and Tzanetakis [34], and Hörschläger et al. [28]. As a reference for non-academic algorithms we evaluate tempo estimators shipped with popular DJ tools, namely *Cross DJ Free*,¹¹ *Deckadance v2 (trial)*,¹² *Traktor 2 PRO*,¹³ and *Rekordbox v3.2.2*.¹⁴ We argue that those estimators are tailored to EDM and therefore should be able to perform well on this data set.

The commercial products typically enable (or require) the user to set an output range for BPM prediction, as a means of dealing with octave errors. *Deckadance* offers to choose among a predefined set of lower bounds of which we selected 80 BPM. In the *Traktor* option pane, the user can choose between a predefined set of tempo ranges. We decided to evaluate two ranges: 88-175 and 60-200 BPM. Similarly, for *CrossDJ*, we chose the 75-150 BPM setting as this is the best match for the given BPM distribution. The research algorithm by Böck et al. [4] also allows to set an arbitrary range. To compare to some of the range presets in commercial products, we evaluate the ranges 50-240, 95-190, and 88-175 BPM.

5.1.2 Results

Table 4 reports the obtained tempo accuracy values for all algorithms. As expected, commercial products outperform research algorithms, however none of the approaches exceeds 77% in terms of accuracy 1. One important finding of more detailed investigations on a per-style level is that the proper choice of the output tempo range has a considerable influence on the accuracy for the style drum-and-bass. For instance, the algorithm by Böck et al. has a known deficiency when dealing with syncopated beats, thus, yielding only acceptable performance on drum-and-bass when being restricted to the 95-190 BPM range. Due to the fact that drum-and-bass makes up 20.9% of the collection, improvements in this style have a significant impact on the overall accuracy. This is further evidenced by [28], where performance is boosted through style-specific output ranges.

5.2 *GiantSteps* Key Data Set

The evaluation method follows the MIREX standard in key estimation tests. It assigns different weighting factors to different types of errors, depending of the proximity of the estimated key to the ground truth (fifth, relative, or parallel keys), and an overall weighted score.¹⁵

¹¹ <http://www.mixvibes.com/products/cross>

¹² <http://www.image-line.com/deckadance/>

¹³ <http://www.native-instruments.com/products/traktor/dj-software/traktor-pro-2/>

¹⁴ <http://rekordbox.com>

¹⁵ http://www.music-ir.org/mirex/wiki/2015:Audio_Key_Detection

	accuracy1	accuracy2
Beatport	4.819	23.795
Davies, Plumbley [14]	29.367	48.042
Böck et al. [4] (50-240)	56.325	88.253
Böck et al. [4] (95-190)	76.506	86.597
Böck et al. [4] (88-175)	69.289	85.693
Gkiokas et al. [21]	58.886	82.380
Percival, Tzanetakis [34]	51.355	88.404
Hörschläger et al. [28]	75.000	82.831
Deckadance (80+)	57.681	81.627
CrossDJ (75-150)	63.404	90.211
Traktor (60-200)	64.608	88.705
Traktor (88-175)	76.958	88.705
Rekordbox	74.548	89.157

Table 4. Tempo estimation accuracies within a $\pm 4\%$ window for evaluated algorithms. BPM range restrictions in parentheses, if applicable.

5.2.1 Algorithms

On top of the *Beatport* annotations that serve as a baseline, we evaluate five different key estimation algorithms: two academic algorithms, namely *Queen Mary’s* Key Detector (QM-Key) [9] and *UPF’s* *Essentia* key extractor [5], and three popular solutions, namely *KeyFinder*,¹⁶ an open-source application by Sha’ath [36], the commercial software *Mixed-In-Key* 7,¹⁷ and the online service/app *Rekordbox* v3.2.2. These applications are regarded trustworthy options for key detection within the EDM community.

KeyFinder is an application that allows the user to tweak the parameters of the algorithm, providing a single estimate per track. We use the default settings. On the other hand *Mixed-In-Key* 7 and *Rekordbox* have a sealed approach and do not give the user any configuration option.

5.2.2 Results

Table 5 shows the results of the different algorithms on the key data set. If we look at the *Beatport* annotations, less than a third of the annotations match the ground truth (29.1%). However, it should be recalled that the majority of the collection (388 tracks) has been collected from reported mistakes in the *Beatport* forum, so the amount of correct keys is consequently very low.¹⁸

From the algorithms in the evaluation, we observe that the two academic algorithms perform poorly on this repertoire, very close to the baseline provided by the *Beatport* key tags, especially *Essentia*.

The two undisclosed approaches yield the best results, with *Rekordbox* providing 71.85% of correct estimations and a weighted score of 79.55 points. In any case, the experiment shows that there is room for improvement of the task in this specific repertoire.

¹⁶ <http://www.ibrahimshaath.co.uk/keyfinder/>

¹⁷ <http://www.mixedinkey.com/>

¹⁸ As a matter of fact, if we look at the performance of the *Beatport* algorithm on the different sources of ground truth separately, we find that the tracks from the user forum only contain 4.2% of correct predictions, while the manually-annotated expert sources result in about 66% of correct predictions each.

	corr.	5th	rel.	par.	other	weigh.
Beatport	29.14	21.52	8.77	19.20	21.36	46.37
QM-Key [9]	39.40	16.89	13.41	5.13	25.17	52.90
Essentia [5]	30.46	17.55	11.09	11.42	29.47	44.85
KeyFinder [36]	45.36	20.69	6.79	7.78	19.37	59.30
Mixed-In-Key	67.22	9.27	5.63	5.30	12.58	74.60
Rekordbox	71.85	10.10	3.97	7.28	6.79	79.55

Table 5. MIREX-style scores on the Key set with results from different algorithms.

6. CONCLUSIONS

We have presented two new data sets for tempo and key estimation in electronic dance music with 664 and 604 examples, respectively. The annotations have been automatically extracted from human feedback. In order to confirm the correctness of the labels, we have inspected randomly selected 15% of the annotations manually and found them all to be correct. In order to make this data set available to the community, we offer the annotations for download on a dedicated web page alongside scripts to retrieve the corresponding audio files from *Beatport* (and a backup location in case files change or are removed) and the original data including the crawl from the user forum and the code to extract the ground truth.¹⁹ Since we performed rather restrictive filtering, a semi-automatic approach, for instance, would allow to extract even more ground truth labels for future work.

From the benchmarking results, we can see that there is still room for improvement for MIR algorithms. Although the data set is biased towards examples that are hard to classify specifically for the *Beatport* algorithms, these results challenge the stereotypical view on EDM as being “trivial cases”. Commercial algorithms are ahead of research-oriented multi-purpose algorithms for both tempo and key estimation as they are likely optimized for EDM. We can conclude that academic algorithms still need to be improved in order to meet the characteristics of EDM, something we wish to contribute to with the publication of these new data sets.

7. ACKNOWLEDGMENTS

This work is supported by the European Union Seventh Framework Programme FP7 / 2007-2013 through the GiantSteps project (grant agreement no. 610591).

8. REFERENCES

- [1] A. Aljanaki, M. Soleymani, F. Wiering, and R. Veltkamp. Mediaeval 2014: A multimodal approach to drop detection in electronic dance music. In *Proc MediaEval Workshop*, 2014.
- [2] A. Barbancho, I. Barbancho, L. Tardón, and E. Molina. *Database of Piano Chords: An Engineering View of Harmony*. Springer, 2013.
- [3] T. Bertin-Mahieux, D. Ellis, B. Whitman, and P. Lamere. The Million Song Dataset. In *Proc 12th ISMIR*, 2011.

¹⁹ <http://www.cpk.jku.at/datasets/giantsteps/>

- [4] S. Böck, F. Krebs, and G. Widmer. Accurate tempo estimation based on recurrent neural networks and resonating comb filters. In *Proc 16th ISMIR*, 2015.
- [5] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra. *Essentia*: an open-source library for sound and music analysis. In *Proc 21st ACM Multimedia*, 2013.
- [6] B. Brewster and F. Broughton. *Last Night a DJ Saved My Life: The History of the Disc Jockey*. Grove/Atlantic, 2007.
- [7] J. Burgoyne, J. Wild, and I. Fujinaga. An expert ground-truth set for audio chord recognition and music analysis. In *Proc 12th ISMIR*, 2011.
- [8] M. Butler. *Unlocking the Groove: Rhythm, Meter, and Musical Design in Electronic Dance Music*. Profiles in popular music. Indiana University Press, 2006.
- [9] C. Cannam, M. Mauch, M. Davies, S. Dixon, C. Landone, K. Noland, M. Levy, M. Zanoni, D. Stowell, and L. Figueira. MIREX 2013 entry: Vamp plugins from the Centre for Digital Music, 2013.
- [10] N. Collins. Influence in early electronic dance music: An audio content analysis investigation. In *Proc 13th ISMIR*, 2012.
- [11] N. Collins. The UbuWeb electronic music corpus: an MIR investigation of a historical database. *Organised Sound*, 20(1), 2015.
- [12] N. Collins and A. McLean. Algorave: A survey of the history, aesthetics and technology of live performance of algorithmic electronic dance music. In *Proc 14th NIME*, 2014.
- [13] M. Davies, N. Degara, and M. Plumbley. Evaluation methods for musical audio beat tracking algorithms. Tech Report C4DM-TR-09-06, Queen Mary University of London, Centre for Digital Music, 2009.
- [14] M. Davies and M. Plumbley. Context-dependent beat tracking of musical audio. *IEEE TASLP*, 15, 2007.
- [15] T. de Clercq and D. Temperley. A corpus analysis of rock harmony. *Popular Music*, 30, 2011.
- [16] B. di Giorgi, M. Zanoni, A. Sarti, and S. Tubaro. Automatic chord recognition based on the probabilistic modeling of diatonic modal harmony. In *Proc 8th Int'l Workshop on Multidimensional Systems (nDS)*, 2013.
- [17] D. Diakopoulos, O. Vallis, J. Hochenbaum, J. Murphy, and A. Kapur. 21st century electronica: MIR techniques for classification and performance. In *Proc 10th ISMIR*, 2009.
- [18] A. Eigenfeldt and P. Pasquier. Evolving structures for electronic dance music. In *Proc 15th Conf on Genetic and Evolutionary Computation (GECCO)*, 2013.
- [19] L.-M. Garcia. On and on: Repetition as process and pleasure in electronic dance music. *Music Theory Online*, 11(4), 2005.
- [20] D. Gärtner. Tempo detection of urban music using tatum grid non negative matrix factorization. In *Proc 14th ISMIR*, 2013.
- [21] A. Gkiokas, V. Katsouros, G. Carayannis, and T. Stafylakis. Music tempo estimation and beat tracking by applying source separation and metrical relations. In *Proc 37th ICASSP*, 2012.
- [22] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: Popular, classical, and jazz music databases. In *Proc 3rd ISMIR*, 2002.
- [23] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE TASLP*, 14(5), 2006.
- [24] S. Hainsworth and M. Macleod. Particle filtering applied to musical tempo tracking. *EURASIP JASP*, 2004:15, 2004.
- [25] C. Harte. *Towards Automatic Extraction of Harmony Information from Music Signals*. PhD thesis, Queen Mary University of London, 2010.
- [26] J. Hockman, M. Davies, and I. Fujinaga. One in the jungle: Downbeat detection in hardcore, jungle, and drum and bass. In *Proc 13th ISMIR*, 2012.
- [27] A. Holzapfel, M. Davies, J. Zapata, J. Oliveira, and F. Gouyon. Selective sampling for beat tracking evaluation. *IEEE TASLP*, 20(9), 2012.
- [28] F. Hörschläger, R. Vogl, S. Böck, and P. Knees. Addressing tempo estimation octave errors in electronic music by incorporating style information extracted from Wikipedia. In *Proc 12th SMC*, 2015.
- [29] K. Jacobson, M. Davies, and M. Sandler. Toward textual annotation of rhythmic style in electronic dance music. In *AES Conv 123*, 2007.
- [30] T. Kell and G. Tzanetakis. Empirical analysis of track selection and ordering in electronic dance music using audio feature extraction. In *Proc 14th ISMIR*, 2013.
- [31] M. Leimeister, D. Gärtner, and C. Dittmar. Rhythmic classification of electronic dance music. In *AES 53rd Int'l Conf: Semantic Audio*, 2014.
- [32] M. Mauch, C. Cannam, M. Davies, C. Harte, S. Kolozali, D. Tidhar, and M. Sandler. Omas2 metadata project 2009. 10th ISMIR Late-Breaking Session, 2009.
- [33] M. Panteli, N. Bogaards, and A. Honingh. Modeling rhythm similarity for electronic dance music. In *Proc 15th ISMIR*, 2014.
- [34] G. Percival and G. Tzanetakis. Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses. *IEEE/ACM TASLP*, 22(12), 2014.
- [35] B. Rocha, N. Bogaards, and A. Honingh. Segmentation and timbre- and rhythm similarity in electronic dance music. In *Proc 10th SMC*, 2013.
- [36] I. Sha'ath. Estimation of key in digital music recordings. Tech Report, Birkbeck College, University of London, 2011.
- [37] M. Sordo, J. Serrà, G. Koduri, and X. Serra. Extracting semantic information from an online carnatic music forum. In *Proc 13th ISMIR*, 2012.
- [38] B. Sturm. Classification accuracy is not enough: On the evaluation of music genre recognition systems. *JIS*, 41(3), 2013.
- [39] B. Sturm. The GTZAN dataset: Its contents, its faults, their effects on evaluation, and its future use. *arXiv.org (e-prints)*, 2013.
- [40] P. Tagg. Debate: From refrain to rave: the decline of figure and the rise of ground. *Popular Music*, 13, 1994.
- [41] R. Wooller and A. Brown. A framework for discussing tonality in electronic dance music. In *Sound: Space - Proc Australasian Computer Music Conference*, 2008.
- [42] R. Wooller and A. Brown. Note sequence morphing algorithms for performance of electronic dance music. *Digital Creativity*, 22(1), 2011.
- [43] K. Yadati, M. Larson, C. Liem, and A. Hanjalic. Detecting drops in electronic dance music: Content-based approaches to a socially significant music event. In *Proc 15th ISMIR*, 2014.
- [44] H. Zeiner-Henriksen. Moved by the groove: Bass drum sounds and body movements in electronic dance music. In Danielsen, ed. *Musical Rhythm in the Age of Digital Reproduction*. Ashgate, Farnham, Surrey, UK, 2010.

TOWARDS SUPPORT FOR UNDERSTANDING CLASSICAL MUSIC: ALIGNMENT OF CONTENT DESCRIPTIONS ON THE WEB

Taku Kuribayashi* Yasuhito Asano Masatoshi Yoshikawa

Graduate School of Informatics, Kyoto University, Japan

choco.ms@gmail.com, {asano, yoshikawa}@i.kyoto-u.ac.jp

ABSTRACT

Supporting the understanding of classical music is an important topic that involves various research fields such as text analysis and acoustics analysis. Content descriptions are explanations of classical music compositions that help a person to understand technical aspects of the music. Recently, Kuribayashi et al. proposed a method for obtaining content descriptions from the web. However, the content descriptions on a single page frequently explain a specific part of a composition only. Therefore, a person who wants to fully understand the composition suffers from a time-consuming task, which seems almost impossible for a novice of classical music. To integrate the content descriptions obtained from multiple pages, we propose a method for aligning each pair of paragraphs of such descriptions. Using dynamic time warping-based method along with our new ideas, (a) a distribution-based distance measure named *w2DD*, and (b) the concept of *passage expressions*, it is possible to align content descriptions of classical music better than when using cutting-edge text analysis methods. Our method can be extended in future studies to create applications systems to integrate descriptions with musical scores and performances.

1. INTRODUCTION

When listening to classical music, we can enhance our understanding of the music by reading descriptions of the contents of the music simultaneously, which is even truer for those who are not experts of the field of music, such as amateur players in a college orchestra. Those people would want to read content descriptions written by experts when they play or listen to a composition.

A content description of classical music is defined as an objective description related to the structure of the composition that explains specific parts of it, often using technical terms and the names of instruments [19]. Reading those passages along with the music can help people to understand what the part we are listening to means technically,



© Taku Kuribayashi, Yasuhito Asano, Masatoshi Yoshikawa. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Taku Kuribayashi, Yasuhito Asano, Masatoshi Yoshikawa. "Towards Support for Understanding Classical Music: Alignment of Content Descriptions on the Web", 16th International Society for Music Information Retrieval Conference, 2015.

which is difficult to understand without preliminary knowledge. An example of a content description of Beethoven's Symphony No. 9,¹ is the following. "The opening theme, played pianissimo over string tremolos, so much resembles the sound of an orchestra tuning, many commentators have suggested that it was Beethoven's inspiration." This example of a content description explains what instruments (strings) are doing technically (pianissimo, tremolos) in a specific part (the opening theme).

Books and the web are two major sources of content descriptions of classical music. Any person having such an interest can find important musical knowledge by reading books such as the well-known *A History of Western Music* [14], which includes not only historical knowledge of the development of western music but also abundant references to other important books. Some encyclopedias contain descriptions of orchestral compositions.

Nevertheless, books have several important limitations. One is that they can hold only a few descriptions. Another problem is that once books are published, they cannot be updated easily or consistently. Although classical music compositions are not increasing to any great degree, the performances are increasing constantly. With the rise of the internet and international communication, there are more descriptions of music and performances. Different perspectives and ways of analysis continue to appear. With the form of printed publications, it is difficult to update the increasing amount of information continuously.

The web is an alternative source of information, offering resources such as "DW3 Classical Music Resources" [11] or Wikipedia. However, it is often difficult to find sufficient information to understand some compositions. Conventional search engines are unsuitable for the vertical search for content descriptions because their results often include commercial websites that do not describe the contents of the compositions. Kuribayashi et al. [19] proposed a method that we can use to collect descriptions from the web. Content descriptions gathered from a number of web pages using their method can be classified into two categories: ones that describe the overall contents of the music, and ones that describe specific parts of the composition. We call the latter ones *partial content descriptions*. Both are essential for technical understanding of the music, although it is often difficult to understand where in the composition partial content descriptions explain. Furthermore,

¹[http://en.wikipedia.org/wiki/Symphony_No._9_\(Beethoven\)](http://en.wikipedia.org/wiki/Symphony_No._9_(Beethoven)), viewed on Jan. 4, 2013

*Current Affiliation: Accenture Japan Ltd.

a single page seldom includes partial content descriptions explains every important part of the composition; a page might describe the introduction in detail, while another page might explain the final part mainly. Therefore, it can be helpful to integrate pieces of information in partial content descriptions from different sources, that is, to check how they complement each other.

We propose a method for the alignment of every pair of paragraphs which are partial content descriptions in different web pages. As a dataset, we manually extract paragraphs corresponding to partial content descriptions from the content descriptions collected from multiple web pages by the method of Kuribayashi et al. [19]. Each alignment clarifies which sentence in a paragraph matches a sentence in the other paragraph. We can understand the music more easily and efficiently by seeing the alignments which integrate the pieces of information in them than merely by reading a single web page. Actually, showing the alignment is beneficial in many situations, as for (1) beginners, (2) experts who want to support those beginners, and (3) future applications.

(1) For beginners, the alignment can help them integrate pieces of information from different websites. If beginners have difficulties understanding one website or feel the need for more information related to a specific part of the composition, they can look at the information that corresponds to the specified part of the description.

(2) For those with a specialized knowledge of classical music, there is always a demand that they want to support beginners as they come to understand music. Web services such as YouTube have several videos that are designed to help beginners to understand classical music. However, preparing all the materials necessary for the explanation is a task that is both difficult and time-consuming. Showing the alignment of sentences provides materials that can support greater understanding. Therefore, showing such an alignment is an important aid to experts who try to support beginners.

(3) In the future, we seek to develop a system that integrates our methods and studies of the analysis of music and music scores; the most important feature is to align content descriptions with the music itself. Beginners will especially benefit from this system because the hardest task for beginners is to ascertain which part of the music the partial content descriptions are referring to. The first step of this ultimate application is analysis of the sentences and their mutual alignment.

The main contributions of this paper are as follows.

- We proposed a novel method named w2DD+PE for aligning partial content descriptions based on dynamic time warping using the following two ideas: (a) the distribution divergence of semantic vectors of words, and (b) *passage expressions*.
- We presented a way to show the aggregated results of our methods for collecting and aligning partial content descriptions.

2. RELATED WORK

This paper deals with various fields of study, including analysis of music, temporal information, multi-document summarization, and parallel corpus discovery; the subject of this paper is the analysis of temporal information in music, and the methodology utilizes the ideas of multi-document summarization and parallel corpus discover models. We will take a look at some of the previous works related to each field of study.

2.1 Musical Knowledge

Music has remained an important topic of research from various aspects, including acoustics, music theory, and psychology. We list a few related works that are closely related to understanding the support and analysis of music.

In the area of understanding support and collecting musical knowledge, Fineman [11] reported a project called “DW3 Classical Music Resources.” The project was a collection of web links that gathered various forms of knowledge related to classical music for college students majoring in music. The link quality was scrutinized by experts, making it easier for students to obtain information that cannot be found easily via conventional web searches. Unfortunately, the project was ceased in 2007.

Other works that are related to the future application of this research include the following. Some studies have been made to analyze the structure of the music itself, such as research by Sumi et al. [36], which created a system for inference of the chord from other data such as the base pitch. Maezawa et al. [24] proposed a system that links the performance and the interpretation of the composition. Using these studies with our research, it would be possible in the future to analyze and extract the music structure and link it to the content descriptions of the composition.

2.2 Temporal Information

As Alonso et al. state in [2], temporal information is an important factor in information retrieval in general. Researches that deal with temporal information in natural language are being studied widely, as in [34], [27], [23], [3], [16]. In order to extract temporal expression, Schilder et al. [32] use finite state transducer (FST); Strötgen et al. [33] use regular expressions, and Mani et al. [26] use machine learning. Chambers et al. [7] focus on the relationships between events, whereas Lapata et al. [20] concentrate on the relationships between expressions in a single sentence. Kimura et al. [17] propose a system that shows chronologically organized information obtained by web searching on a single person. Schilder et al. [32] extract temporal information from news articles.

As we see from these examples, researches on temporal information have various aspects, including many viewpoints on the subject and granularity. In our research, we deal with temporal information in one composition, which is generally an hour or two at the longest.

2.3 Multi-document Summarization

To gain knowledge from multiple sources, summarization of information is an important technique. One type of summarization, the extractive method, chooses subsets of the original document to convey the meaning of the whole text. In the task of multiple document summarization, numerous approaches have been taken. Mani et al. [25] take a graph-based approach, and many of recent studies follow similar ideas [39] [8] [5] [13] [10]. Other approaches include Bayesian models [9] [6], topic models [15], rhetoric-based models [4], and cluster-based models [30] [37].

2.4 Parallel Corpus Discovery

Because we are interested in discovering potential alignments from different documents, we will take a look at previous works that utilize techniques for investigating parallel corpora. Caroline et al. [21] apply dynamic time warping to movie subtitles to construct parallel corpora for machine translation. Previous works on finding parallel texts from bilingual, often non-parallel, corpora include [12], [29], [38] and [35].

3. ALIGNMENT OF DESCRIPTIONS

3.1 Collection of Content Descriptions

We adopted a method of Kuribayashi et al. [19] to collect content descriptions from the web. Their method utilizes labeled latent Dirichlet allocation (labeled LDA) [31] which is a supervised learning to classify documents probabilistically. They proposed eight classes of descriptions (one of them corresponds to content descriptions) contained in the pages obtained by inputting names of compositions to a search engine, and trained labeled LDA with manually-classified 1540 pages. Note that information other than text, such as images or HTML tags, is removed from these pages using nwc-toolkit². Applying the trained labeled LDA to paragraphs obtained by inputting the name of a composition to a search engine, we can collect paragraphs corresponding to content descriptions.

From our investigation of a number of content descriptions gathered by this method, we found out that partial content descriptions in a single paragraph are ordered chronologically for a composition. Therefore, the sequence alignment of those sentences is more suitable for integrating information of partial content descriptions than other methods such as matching sentences.

3.2 Bootstrapping Method for Acquiring Passage Expressions

It is quite a difficult task to identify what part of a musical piece that a description corresponds, because most partial content descriptions do not contain measure numbers. Here we try to obtain as much information related to the correspondence of one expression to another. For instance, if we have two descriptions “The first theme is played by

solo flute” and “The lyrical first subject appears after the introduction,” we can see the relationship between them; because the words “theme” and “subject” are semantically similar in this context, we can align these two sentences and understand that the theme is lyrical and is played by the flute. If we have another sentence talking about “solo flute,” we can also infer the relationship of that description to the two sentences above. We have to identify what types of nouns point to the parts of music, which we call *passage expressions*, in order to perform this inference. If we are able to obtain those expressions, then we would be able to use them to align sentences that correspond to the same part of the composition. In the future, we might also be able to employ them to mapping of the actual parts of music by finding measure numbers or giving some information manually.

To obtain the passage expressions, we focus on the grammatical structure of content description sentences. In content descriptions, the most basic structure of sentences is subject-verb-object, where the verb describes the relationship between two passage expressions (subject and object). Therefore, we use a bootstrapping method as in [1], using the relation between the subject and the predicate to extract appropriate nouns. Because a simple bootstrapping method tends to produce noises in the results, we also proposed filtering methods to reduce those noises.

The corpus for the bootstrapping method is 2300 paragraphs which are the top 100 paragraphs obtained by applying the method of Kuribayashi et al. [19] to each of 23 compositions.

First, we prepared an initial list of 14 nouns and 29 verbs for the bootstrapping method. Then we expanded that list when two of the triplet of the subject, the verb and the object (or the object of the preposition) were already in the list, by adding the third word. We did not add the third word when the subject was a personal pronoun (“I”, “we”, “you”, “he”, or “she”) because the word was inappropriate in almost all cases.

Instead of adding all the words that appear in the triplet, we eliminate words that do not fulfill certain conditions to reduce noise words that are not relevant to content descriptions. The following filtering methods incorporate the results of the labeled LDA-based method [19] and word2vec [28]³ which converts a word to a vector based on the occurrence of words in a corpus; the similarity of words can be calculated using the vectors corresponding to the words.

L-LDA Words that are stop words or that do not appear in the training data of labeled LDA in the methods of [19] are not added.

word2vec Words that are below the threshold (0.128 and 0.3) of word2vec similarity. Word2vec using the same corpus as the one used for our bootstrapping explained above. The word2vec similarity used here is defined as the maximum of the similarities between the word and the seed nouns of the bootstrapping method.

²<http://nwc-toolkit.googlecode.com/svn/trunk/docs/tools/text-extractor.html>

³<https://code.google.com/p/word2vec/>

L-LDA && word2vec Only words that fulfill both of the above two are added. The threshold of the word2vec score is 0.128.

L-LDA || word2vec Words that fulfill either one of the two above are added. The threshold of the word2vec score is 0.3.

3.3 Alignment Method using Dynamic Time Warping

We propose a method called “word sets to Distribution Distance-based alignment using Passage Expressions” (w2DD +PE) for finding an alignment of pairs of paragraphs of content descriptions. This method is based on dynamic time warping (DTW), a well-known technique for finding an alignment of two sequences. Applying DTW to paragraphs requires a distance measure of two sentences. We propose a new distance measure employing (a) the distribution of word vectors of each sentence, and (b) passage expressions.

3.3.1 w2DD

A simple measure of distance between two sentences is to take the average of semantic vectors of words in the sentence calculated using word2vec and calculate the cosine distance. However, adopting the average loses much information about how the vectors distribute. Therefore, it is required to propose a new method to capture the feature of the distribution corresponding to each sentence.

The fundamental idea of our new method, named w2DD, is to measure the distance between two sentences by the distance between the distributions of their corresponding vectors. We firstly reduce each vector to a small number of dimensions using principal component analysis because the 200 dimensions obtained by word2vec are too numerous to handle. The number of dimensions is determined empirically, and eleven dimensions were sufficient for the cumulative proportion of 70%. Secondly, we convert the 11-dimension vectors of each sentence into a histogram, in order to apply a distance measure for a pair of probabilistic distributions. For the conversion, we divide each dimension into halves (resulting in 2^{11} subspaces) and count the number of vectors in each subspace; the sequence of the numbers forms the obtained histogram. We tried splitting each dimension into 2, 3, and 4, but the result did not change at all, so we chose 2. Then we calculated the distance of the pair of histograms by the Jensen–Shannon divergence using the following formula:

$$JSD(P \| Q) = \frac{1}{2} \left(\sum_x \log \frac{P(x)}{R(x)} + \sum_x \log \frac{Q(x)}{R(x)} \right) \quad (1)$$

where P and Q are the histograms corresponding to the two sets of vectors, x is each subspace, $P(x)$ is the number of vectors of P in x divided by the total number of vectors of P , and $R(x) = \frac{P(x)+Q(x)}{2}$.

3.3.2 Passage Expressions

First, to utilize the information of passage expressions in sentences containing no such expressions, we merge such

sentences into the previous sentence having a passage expression. Then, we calculate the distance of two sentences s_1 and s_2 as follows. Let $\text{sim}(p_1, p_2)$ be the cosine similarity between the semantic vectors of passage expressions p_1 in s_1 and p_2 in s_2 . The distance $\text{Dist}(s_1, s_2)$ is

$$\text{Dist}(s_1, s_2) = \begin{cases} \alpha JSD(s_1, s_2) \\ \quad + (1 - \alpha)(1 - \max_{p_1, p_2}(\text{sim}(p_1, p_2))) \\ \quad \text{(if } \max_{p_1, p_2}(\text{sim}(p_1, p_2)) \neq 0) \\ JSD(s_1, s_2) \quad \text{(otherwise)} \end{cases} \quad (2)$$

where $JSD(s_1, s_2)$ is the value calculated as in Section 3.3.1. If either one of the paragraphs is without a passage expression, then $\max_{p_1, p_2}(\text{sim}(p_1, p_2)) = 0$. Therefore only the Jensen–Shannon divergence matters. Also, α is the coefficient factor, which was set to 0.2, 0.4, 0.6, 0.8, and 1.0.

4. EVALUATION

4.1 Procedure

An input of the alignment is a pair of paragraphs which are partial content descriptions explaining a common section in a composition. The labeled LDA-based method of Kuribayashi et al. [19] is able to collect content descriptions, although it is not able to extract partial content descriptions from them. Consequently, our data set consists of 32 paragraphs (135 sentences) manually extracted from the top 100 paragraphs for each of 10 classical music compositions obtained by their method; the number of pairs are 41. The extraction and assignment of each paragraph to a section is based on keywords corresponding to sections, such as “movement” (the most basic divisions of a music composition), “exposition” and “development” (common structures within a movement). The keywords are selected for the sonata form, and a selection specialized for other types of classic music, “theme and variations” for example, is also possible. A method for automatic extraction and assignment is a candidate of future studies.

To see how each of our ideas work, we used the following variants of methods for calculating the distance between two sentences.

Baseline1 the cosine distance of the averages of word2vec vectors.

Baseline1+PE the cosine distance of the following 400 dimension vectors for the two sentences s_1 and s_2 . The first 200 dimensions are the average of the word2vec vector of each sentence. The second 200 dimensions are the word2vec vector of passage expression p_1 for s_1 or p_2 for s_2 , respectively; p_1 and p_2 are the pair of the closest expressions in terms of word2vec cosine similarity.

Baseline2 the cosine distance calculated using sentence2vec.⁴ This is an implementation of *Paragraph Vector* proposed by Le and Mikolov [22],

⁴<https://github.com/klb3713/sentence2vec>

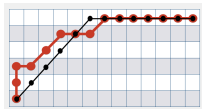


Figure 1. Example of how to calculate F -measure.

which is an advanced method of word2vec that incorporates the order of words in a sentence to represent its semantics. Their experiments showed that *Paragraph Vector* performs better than previous methods for several tasks; word vector averaging, Naive Bayes, SVMs, and recursive neural network for a sentiment analysis task; vector averaging, bag-of-words, and bag-of-bigrams for an information retrieval task.

Baseline2+PE the cosine distance calculated using sentence2vec incorporated with the passage expression vector by the same way Section 3.3.2.

w2DD the method described in Section 3.3.1

w2DD+PE the method described in Section 3.3.2.

For Baseline1+PE, Baseline2+PE, and w2DD+PE, we used the filtered list of passage expressions described in Section 3.1.

The ground truth for the alignment of each pair of paragraphs was created manually by one of the authors who is an enthusiast of classical music, with the help of various books and websites on the compositions. We evaluate each method using precision, recall, and F -measure. We explain below how they are calculated employing Figure 1 which illustrates an example of the result of alignment of two paragraphs. The red dots represent the manual alignment result, and black dots indicate the output a method; in the manual alignment, the first sentence of paragraph 1 (x -axis) corresponds to the first, second, and third sentences of paragraph 2 (y -axis), the second sentence of paragraph 1 corresponds to the third sentence of paragraph2, and so on. The precision is the number of matching red and black dots over the number of black dots, $9/13$ in this case. The recall is the number of matching red and black dots over the number of red dots, $9/15$ in this case. The F -measure is defined by the following equation.

$$F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (3)$$

4.2 Results

Tables 1, 2, and 3 present the experimental results. Comparing the “No PE” row with the others in each table, we see that employing PE improves the results in general. Comparing the three tables, we see that w2DD performs much better than baseline methods. Especially, w2DD+PE with L-LDA ($\alpha = 0.2$) and w2DD+PE with L-LDA && word2vec ($\alpha = 0.2$) are the methods that resulted in the best F -measure (shown in bold in the Table 3).

Table 1. Results of Baseline1 (No PE) and Baseline1+PE.

Method	Precision	Recall	F -measure
No PE	0.595	0.534	0.563
No Filtering	0.608	0.633	0.620
L-LDA	0.618	0.647	0.632
word2vec (0.128)	0.582	0.618	0.600
word2vec (0.3)	0.562	0.607	0.584
L-LDA && word2vec	0.592	0.629	0.610
L-LDA word2vec	0.591	0.615	0.602

4th mv: Recapitulation	A new theme briefly appears, and the recapitulation begins. The fate theme reappears briefly and the orchestra plays majestic chords in B major, followed by a fermata rest, which gives the impression to the ear that the symphony has reached its end. This has caused more than one audience to erupt in applause, but it is but the end of the recapitulation. Toward the end of the recapitulation, Fate reappears, now just as a rhythm. This leads to an exciting and suspenseful buildup, whose tensions are relaxed when the Fate theme marches forward in its most triumphant form in major, fortissimo, broad, majestic. The moment of suspense just before this grand arrival has turned out to be a ferocious audience trap. People who haveven really been listening but have noticed that the music has stopped are liable to begin their applause at this point. After the Fate theme has made its splendid entrance, the music moves forward into a headlong Presto, broadening again for the closing final pages. The re-transition to recapitulation is rather abrupt, yet a clever use of common tone modulation can be observed.
---------------------------	--

Figure 2. Visualization of Tchaikovsky’s Symphony No.5 (Each block of sentences separated by dotted lines is from a single web page.)

Because the baseline methods “compress” the word vectors in a sentence into a single vector, they are considered to lose much information of the words. On the other hand, w2DD keeps the information on how varied the words are in the sentence.

The numbers of passage expressions employed in L-LDA and L-LDA && word2vec were 30 and 26, respectively. Passage expressions were generally effective as mentioned above, while higher alpha often made the performance worse. These results would indicate that the word distribution employed in w2DD is more important than passage expressions.

4.3 Visualization

To present the results of our methods to users for understanding support, we created a prototype of a system to visualize them in a table form, whose examples can be accessed online.⁵ Each row corresponds to a part of music. Figure 2 shows a single row corresponding to the “4th movement” of the table for Tchaikovsky’s “Symphony No.5.” In this row, there are three blocks of sentences separated by dotted lines, each of which indicates a paragraph retrieved from one web page. As we hover the cursor over one of the sentences, the sentences of other descriptions that are aligned with that sentence by our method is highlighted (shown as pink in the figure).

The recapitulation part builds up the tension and ends

⁵<http://bit.ly/1vHMkgm>

Table 2. Results of Baseline2 (No PE) and Baseline2+PE.

Method	α	Precision	Recall	F -measure
No PE	-	0.610	0.550	0.578
No Filtering	0.2	0.621	0.562	0.590
	0.4	0.630	0.568	0.597
	0.6	0.603	0.544	0.572
	0.8	0.474	0.417	0.444
L-LDA	0.2	0.659	0.598	0.627
	0.4	0.670	0.604	0.635
	0.6	0.627	0.565	0.594
	0.8	0.474	0.417	0.444
word2vec (0.128)	0.2	0.619	0.562	0.589
	0.4	0.630	0.568	0.597
	0.6	0.603	0.544	0.572
	0.8	0.474	0.417	0.444
word2vec (0.3)	0.2	0.625	0.565	0.593
	0.4	0.627	0.565	0.594
	0.6	0.603	0.544	0.572
	0.8	0.474	0.417	0.444
L-LDA && word2vec	0.2	0.659	0.598	0.627
	0.4	0.670	0.604	0.635
	0.6	0.627	0.565	0.594
	0.8	0.474	0.417	0.444
L-LDA word2vec	0.2	0.616	0.559	0.586
	0.4	0.627	0.565	0.594
	0.6	0.603	0.544	0.572
	0.8	0.474	0.417	0.444

up with a brief stop. From the three highlighted descriptions, it is readily apparent that common tone modulation is used cleverly in the recapitulation; the fate theme engenders a suspenseful buildup; and a fermata rest follows the majestic chords in B major. By reading the aligned descriptions that are retrieved from multiple pages, a more detailed and thorough view of the part of the music can be obtained than by reading just one description.

5. CONCLUDING REMARKS

As described in this paper, we proposed methods for supporting the understanding of classical music using mutual alignment of partial content descriptions. Our method w2DD+PE uses word sets to Distribution Distance (w2DD) and the concept of *passage expressions*, which are expressions that serve as the key to identification of which parts of the music the descriptions correspond to. Although the concept of passage expressions is unique to the field of classical music, w2DD can be applied to other domains of text data. It is one of our future tasks to apply w2DD to other datasets.

Future studies will be undertaken to create an application system that can help beginners to appreciate classical music. By integrating our methods with studies of musical analysis such as [36] and [24], or other applications of music-related information retrieval such as [18], it is expected to be possible to support beginners in their efforts to understand and enjoy music.

Table 3. Results of w2DD (No PE) and w2DD+PE.

Method	α	Precision	Recall	F -measure
No PE	-	0.746	0.688	0.716
No Filtering	0.2	0.671	0.733	0.701
	0.4	0.671	0.733	0.701
	0.6	0.690	0.748	0.718
	0.8	0.667	0.718	0.691
L-LDA	0.2	0.680	0.780	0.727
	0.4	0.678	0.774	0.723
	0.6	0.675	0.760	0.715
	0.8	0.669	0.745	0.705
word2vec (0.128)	0.2	0.639	0.721	0.678
	0.4	0.639	0.721	0.678
	0.6	0.651	0.718	0.683
	0.8	0.658	0.718	0.687
word2vec (0.3)	0.2	0.648	0.736	0.689
	0.4	0.648	0.736	0.689
	0.6	0.652	0.739	0.693
	0.8	0.659	0.745	0.699
L-LDA && word2vec	0.2	0.680	0.780	0.727
	0.4	0.678	0.774	0.723
	0.6	0.673	0.757	0.712
	0.8	0.669	0.745	0.705
L-LDA word2vec	0.2	0.637	0.733	0.681
	0.4	0.637	0.733	0.681
	0.6	0.641	0.727	0.681
	0.8	0.651	0.736	0.691

6. ACKNOWLEDGMENTS

This work was supported by JSPS KAKENHI Grant Number 15K00423 and the Kayamori Foundation of Informational Science Advancement.

7. REFERENCES

- [1] E. Agichtein and L. Gravano. Snowball: Extracting relations from large plain-text collections. In *Proc. of the 5th ACM conference on Digital Libraries*, pages 85–94, 2000.
- [2] O. Alonso, M. Gertz, and R. Baeza-Yates. On the value of temporal information in information retrieval. In *ACM SIGIR Forum*, volume 41-2, pages 35–41, 2007.
- [3] O. Alonso, M. Gertz, and R. Baeza-Yates. Clustering and exploring search results using timeline constructions. In *Proc. of the 18th CIKM*, pages 97–106, 2009.
- [4] J. Atkinson and R. Munoz. Rhetorics-based multi-document summarization. *Expert Systems with Applications*, 40(11):4346–4352, 2013.
- [5] E. Canhasi and I. Kononenko. Weighted archetypal analysis of the multi-element graph for query-focused multi-document summarization. *Expert Systems with Applications*, 41(2):535–543, 2014.
- [6] A. Celikyilmaz and D. Hakkani-Tür. Discovery of topically coherent sentences for extractive summarization. In *Proc. of HLT '11*, pages 491–499, 2011.
- [7] N. Chambers, S. Wang, and D. Jurafsky. Classifying temporal relations between events. In *Proc. of the 45th*

- Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, pages 173–176, 2007.
- [8] J. Christensen, Mausam, S. Soderland, and O. Etzioni. Towards Coherent Multi-Document Summarization. In *Proc. of HLT-NAACL '13*, pages 1163–1173, 2013.
- [9] H. Daumé III and D. Marcu. Bayesian query-focused summarization. In *Proc. of the 44th Annual Meeting of the ACL*, pages 305–312, 2006.
- [10] R. Ferreira, L. de Souza Cabral, F. Freitas, R. D. Lins, G. de França Silva, S. J. Simske, and L. Favaro. A multi-document summarization system based on statistics and linguistic treatment. *Expert Systems with Applications*, 41(13):5780–5787, 2014.
- [11] Y. Fineman. DW3 Classical Music Resources: Managing Mozart on the Web. *Libraries and the Academy*, 1(4):383–389, 2001.
- [12] P. Fung and P. Cheung. Mining verynon-parallel corpora: Parallel sentence and lexicon extraction via bootstrapping and em. In *Proc. of EMNLP*, pages 57–63, 2004.
- [13] G. Glavaš and J. Šnajder. Event graphs for information retrieval and multi-document summarization. *Expert Systems with Applications*, 41(15):6904–6916, 2014.
- [14] D. J. Grout, C. V. Palisca, et al. *A History of Western Music*. Number Ed. 5. WW Norton & Company, Inc., 1996.
- [15] A. Haghighi and L. Vanderwende. Exploring content models for multi-document summarization. In *Proc. of the NAACL '09*, pages 362–370, 2009.
- [16] J. Hobbs and J. Pustejovsky. Annotating and reasoning about time and events. In *Proc. of AAAI Spring Symposium on Logical Formalizations of Commonsense Reasoning*, volume 3, pages 74–82, 2003.
- [17] R. Kimura, S. Oyama, H. Toda, and K. Tanaka. Creating personal histories from the Web using namesake disambiguation and event extraction. In *Web Engineering*, pages 400–414, 2007.
- [18] P. Knees, T. Pohle, M. Schedl, and G. Widmer. A music search engine built upon audio-based and web-based similarity measures. In *Proc. of the 30th SIGIR*, pages 447–454, 2007.
- [19] T. Kuribayashi, Y. Asano, and M. Yoshikawa. Ranking method specialized for content descriptions of classical music. In *Poster Proc. of the 22nd WWW*, pages 141–142, 2013.
- [20] M. Lapata and A. Lascarides. Learning Sentence-internal Temporal Relations. *Journal of Artificial Intelligence Research (JAIR)*, 27:85–117, 2006.
- [21] C. Lavecchia, K. Smali, and D. Langlois. Building parallel corpora from movies. In *Proc. of the 4th NLPCS*, 2007.
- [22] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. *Proc. of the 31st ICML*, pages 1188–1196, 2014.
- [23] X. Ling and D. S. Weld. Temporal Information Extraction. In *The AAAI Conference on Artificial Intelligence*, pages 1385–1390, 2010.
- [24] A. Maezawa, M. Goto, and H. G. Okuno. Query-By-Conducting: An interface to retrieve classical-music interpretations by real-time tempo input. In *Proc. of the 11th ISMIR*, pages 477–482, 2010.
- [25] I. Mani and E. Bloedorn. Multi-document summarization by graph search and matching. In *Proc. of AAAI'97/IAAI'97*, pages 622–628, 1997.
- [26] I. Mani, M. Verhagen, B. Wellner, C. Lee, and J. Pustejovsky. Machine learning of temporal relations. In *Proc. of the 44th Annual Meeting of the ACL*, pages 753–760, 2006.
- [27] P. Mazur and R. Dale. Wikiwars: A new corpus for research on temporal expressions. In *Proc. of the EMNLP 2010*, pages 913–922, 2010.
- [28] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. *Proc. of Workshop at ICLR*, 2013.
- [29] D. S. Munteanu and D. Marcu. Extracting parallel sub-sentential fragments from non-parallel corpora. In *Proc. of the 44th Annual Meeting of the ACL*, pages 81–88, 2006.
- [30] D. R. Radev, H. Jing, M. Styś, and D. Tam. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938, 2004.
- [31] D. Ramage, D. Hall, R. Nallapati, and C.D. Manning. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora. In *Proc. of EMNLP, Volume 1*, pages 248–256, 2009.
- [32] F. Schilder and C. Habel. From temporal expressions to temporal information: Semantic tagging of news messages. In *Proc. of the Workshop on Temporal and Spatial Information Processing-Volume 13*, page 9, 2001.
- [33] J. Strötgen and M. Gertz. HeideTime: High quality rule-based extraction and normalization of temporal expressions. In *Proc. of the 5th International Workshop on Semantic Evaluation*, pages 321–324, 2010.
- [34] J. Strötgen and M. Gertz. Multilingual and cross-domain temporal tagging. *Language Resources and Evaluation*, 47(2):269–298, 2013.
- [35] F. Su and B. Babych. Measuring comparability of documents in non-parallel corpora for efficient extraction of (semi-)parallel translation equivalents. In *Proc. of the Joint Workshop on ESIRMT and HyTra*, EACL 2012, pages 10–19, 2012.
- [36] K. Sumi, K. Itoyama, K. Yoshii, K. Komatani, T. Ogata, and H. G. Okuno. Automatic chord recognition based on probabilistic integration of chord transition and bass pitch estimation. In *Proc. of the 9th ISMIR*, pages 39–44, 2008.
- [37] X. Wan and J. Yang. Multi-document summarization using cluster-based link analysis. In *Proc. of SIGIR '08*, pages 299–306, 2008.
- [38] D. Wu and P. Fung. Inversion transduction grammar constraints for mining parallel sentences from quasi-comparable corpora. In R. Dale, K. Wong, J. Su, and O. Kwong, editors, *Natural Language Processing – IJCNLP 2005*, LNCS 3651, pages 257–268, 2005.
- [39] L. Zhao, L. Wu, and X. Huang. Using query expansion in graph-based approach for query-focused multi-document summarization. *Information Processing & Management*, 45(1):35–41, 2009.

FLABASE: TOWARDS THE CREATION OF A FLAMENCO MUSIC KNOWLEDGE BASE

Sergio Oramas¹, Francisco Gómez², Emilia Gómez¹, Joaquín Mora³

¹Music Technology Group, Universitat Pompeu Fabra

²Technical University of Madrid

³Faculty of Psychology, University of Sevilla

{sergio.oramas, emilia.gomez}@upf.edu, fmartin@eui.upm.es, mora@us.es

ABSTRACT

Online information about flamenco music is scattered over different sites and knowledge bases. Unfortunately, there is no common repository that indexes all these data. In this work, information related to flamenco music is gathered from general knowledge bases (e.g., Wikipedia, DBpedia), music encyclopedias (e.g., MusicBrainz), and specialized flamenco websites, and is then integrated into a new knowledge base called FlaBase. As resources from different data sources do not share common identifiers, a process of pair-wise entity resolution has been performed. FlaBase contains information about 1,174 artists, 76 *palos* (flamenco genres), 2,913 albums, 14,078 tracks, and 771 Andalusian locations. It is freely available in RDF and JSON formats. In addition, a method for entity recognition and disambiguation for FlaBase has been created. The system can recognize and disambiguate FlaBase entity references in Spanish texts with an f-measure value of 0.77. We applied it to biographical texts present in FlaBase. By using the extracted information, the knowledge base is populated with relevant information and a semantic graph is created connecting the entities of FlaBase. Artists relevance is then computed over the graph and evaluated according to a flamenco expert criteria. Accuracy of results shows a high degree of quality and completeness of the knowledge base.

1. INTRODUCTION

Music context information is now playing a key role in MIR research. Multimodal approaches, semantic approaches, and text-IR approaches have shown important achievements in typical MIR problems, such as music recommendation and discovery, genre classification, or music similarity [17]. Therefore, collecting and storing music context information may be extremely useful for the MIR research community [13]. There are some broad repositories of music

context information such as MusicBrainz¹ or Discogs². Although some of these repositories are very complete and accurate, there is still a vast amount of music information out there, which is generally scattered among different sources on the Web. Hence, harvesting and combining that information is a crucial step in the creation of practical and meaningful music knowledge bases. In addition, the creation of genre-specific knowledge bases may be very valuable for research and dissemination purposes, and particularly to non-western music traditions.

In this paper, we propose a methodology for the creation of a genre-specific knowledge base; in particular, a knowledge base of flamenco music. The proposed methodology combines content curation and knowledge extraction processes. First, an important amount of information is gathered from different data sources, which are subsequently combined by applying pair-wise entity resolution. Next, new knowledge is extracted from unstructured harvested texts and employed to populate the knowledge base. For this purpose, an entity linking system has been expressly developed. Finally, the content of the knowledge base is used to compute artist relevance and results are evaluated according to flamenco experts criteria. The content of the knowledge base is freely available and downloadable as data dumps in RDF and JSON formats.

The remainder of the paper is organized as follows. In Section 2, an introduction to flamenco music is presented. In Section 3 some relevant prior work is briefly surveyed. Section 4 describes the structure of the knowledge base. Next, in Section 5 the process of content curation is explained. Section 6 shows the methodology applied for knowledge extraction. In Section 7 artist relevance is computed and some statistics about the content are laid out. Finally, Section 8 concludes the paper and points out for future lines of work.

2. FLAMENCO MUSIC

Several musical traditions contributed to the genesis of flamenco music as we know it today. Among them, the influences of the Jews, Arabs, and Spanish folk music are recognizable, but indubitably the imprint of Andalusian Gypsies' culture is deeply ingrained in flamenco music. Fla-



© Sergio Oramas¹, Francisco Gómez², Emilia Gómez¹, Joaquín Mora³.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Sergio Oramas¹, Francisco Gómez², Emilia Gómez¹, Joaquín Mora³. "FlaBase: Towards the Creation of a Flamenco Music Knowledge Base", 16th International Society for Music Information Retrieval Conference, 2015.

¹ <http://musicbrainz.org>

² <http://www.discogs.com/>

menco occurs in a wide range of settings, including festive *juergas* (private parties), *tablaos* (flamenco venues), concerts, and big productions in theaters. In all these settings we find the main components of flamenco music: *cante* or singing, *toque* or guitar playing, and *baile* or dance. According to Gamboa [9], flamenco music grew out of the singing tradition, as a melting process of all the traditions mentioned above, and therefore the role of the singer soon became dominant and fundamental. *Toque* is subordinated to *cante*, especially in more traditional settings, whereas *baile* enjoys more independence from voice.

In the flamenco jargon styles are called *palos*. Criteria adopted to define flamenco *palos* are rhythmic patterns, chord progressions, lyrics and its poetic structure, and geographical origin. In flamenco geographical variation is important to classify *cantes* as often they are associated to a particular region where they were originated or where they are performed with gusto. Rhythm or *compás* is a unique feature of flamenco. Rhythmic patterns based on 12-beat cycles are mainly used. Those patterns can be classed as follows: binary patterns, such as *tangos* or *tientos*; ternary patterns, which are the most common ones, such as *fandangos* or *bulerías*; mixed patterns, where ternary and binary patterns alternate, such as *guajira*; free-form, where there is no a clear underlying rhythm, such as *tonás*. For further information on fundamental aspects of flamenco music, see the book of Fernández [7]. For a comprehensive study of styles, musical forms and history of flamenco the reader is referred to the books of Blas Vega and Ríos Ruiz [3], Navarro and Roperó [12], and Gamboa [9] and the references therein.

3. RELATED WORK

A knowledge base is a centralized repository intended to store both complex structured and unstructured information. Content in a knowledge base can be either curated or extracted, and knowledge bases can be classified according to those criteria [6]. Curated knowledge can be manually gathered by humans or automatically extracted from a structured data source. By contrast, extracted knowledge is produced after the application of an information extraction process over an unstructured data source. There are several well-known general purpose knowledge bases either extracted or curated. The most widely used are DBpedia³ and Freebase⁴, and more recently WikiData⁵. The most relevant extracted knowledge bases are NELL [5] and Open IE [1].

In the music field, one of the most complete and broadly used knowledge bases is MusicBrainz⁶, which has been created in a collaborative curated way. However, there is not any extracted and open music knowledge base. Moreover, little effort have been done in the creation of genre-specific knowledge bases. Most relevant initiatives in this

direction have been done within the CompMusic project⁷. In this project, one of the main tasks has been the gathering of culture-specific corpora of non-western musical traditions, combining expert information, audio recordings, features, music notation, lyrics, editorial metadata and community information [18]. According to [19], a domain-specific corpora should be designed by satisfying the following criteria: purpose, coverage, completeness, quality and reusability. In [15], the architecture and applications of a system that exploits domain-specific corpora is presented. Another interesting project is Linked Jazz [14], where the application of Linked Open Data (LOD) technology to enhance discovery and visibility of jazz music is studied.

4. FLABASE

FlaBase (Flamenco Knowledge Base) is the acronym of a new knowledge base of flamenco music. Its ultimate aim is to gather all available online editorial, biographical and musicological information related to flamenco music. A first version is just being released. Its content is the result of the curation and extraction processes explained in Sections 5 and 6. FlaBase is stored in RDF and JSON formats, and it is freely available for download⁸. Its RDF version follows the Linked Open Data principles, and it might be queried by setting up a SPARQL endpoint. A JSON version is also available, thus facilitating the use of the content by all the community of researchers and developers. This first release of FlaBase contains information about 1,174 artists, 76 *palos* (flamenco genres), 2,913 albums, 14,078 tracks, and 771 Andalusian locations.

4.1 Ontology Definition

The FlaBase data structure is defined in an ontology schema. One of the advantages of using an ontology as a schema is that it can be easily modified. Thus, our design is a first building block that can be enhanced and redefined in the future. The initial ontology is structured around five main classes: MusicArtist, Album, Track, Palo and Place, and three domain specific classes: *cantaor* (flamenco singer), guitarist (flamenco guitar player), and *bailaor* (flamenco dancer). These three classes were defined because they are the most frequent types of artists in the data. Other instrument players may be instantiated directly from the MusicArtist class. We have tried to reuse as much vocabulary as we could. We re-utilized most of the classes and some properties from the Music Ontology⁹, a standard model for publishing music-related data. We selected the classes according to the ones used by the LinkedBrainz project¹⁰, which maps concepts from MusicBrainz to Music Ontology.

³ <http://dbpedia.org>

⁴ <http://www.freebase.com>

⁵ <http://www.wikidata.org>

⁶ <http://musicbrainz.org>

⁷ <http://compmusic.upf.edu>

⁸ <http://mtg.upf.edu/download/datasets/flabase>

⁹ <http://musicontology.com>

¹⁰ <https://wiki.musicbrainz.org/LinkedBrainz>

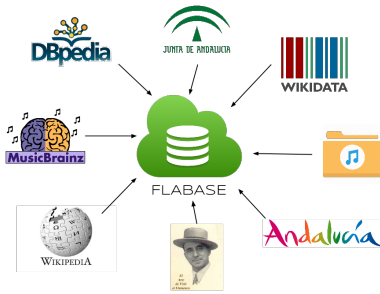


Figure 1. Selected data sources

5. CONTENT CURATION

The first step towards building a domain-specific knowledge base is to gather all possible content from available data sources. This implies at least two problems, namely, the selection of sources, and the matching between entities from different sources. In what follows we enumerate the involved data sources and describe the methodology applied to entity resolution.

5.1 Data Acquisition

Our aim is to gather an important amount of information about musical entities, including textual descriptions and available metadata. A schema of the selected data sources is shown in Figure 1. We started by looking at Wikipedia¹¹, the free and multilingual Internet encyclopedia. It is the Internet's largest and most popular general reference work. Each Wikipedia article may have a set of associated categories. Categories are intended to group together pages on similar subjects and are structured in a taxonomical way. To find Wikipedia articles related to flamenco music, we first looked for flamenco categories. The taxonomy of categories can be explored by querying DBpedia, a knowledge base with structured content extracted from Wikipedia. In particular, we employed the SPARQL endpoint of the Spanish DBpedia¹². We queried for categories related to the flamenco category in the taxonomy. At the end, we obtained 17 different categories (e.g., *cantaoras de flamenco*, *guitarristas de flamenco*).

By querying again DBpedia, we gathered all DBpedia resources related to one of these categories. We obtained a total number of 438 resources in Spanish, of which 281 were also in English. Each DBpedia resource is associated with a Wikipedia article. Text and HTML code were then extracted from Wikipedia articles in English and Spanish by using the WikiMedia API. Next, we classified the extracted articles according to the ontology schema defined in our knowledge base (Section 4.1). For this purpose, we exploited classification information provided by DBpedia

(DBpedia ontology and Wikipedia categories). At the end, from all gathered resources, we only kept those related to artists and *palos*, totalling 291 artists and 56 *palos*.

However, the amount of information present in Wikipedia related to flamenco music is somewhat scarce. Therefore, we decided to expand our knowledge base with information from two different websites. First, *Andaluçia.org*, the touristic web from the Andalusian Government¹³. It contains 422 artist biographies in English and Spanish, and the description of 76 *palos* also in both languages. Second, a website called *El arte de vivir el flamenco*¹⁴, which includes 749 artist biographies among *cantaoras*, *bailaoras* and guitarists. Both webs were crawled and their content stored in our knowledge base.

MusicBrainz is one of the biggest and more reliable open music databases, which provides an unambiguous form of music identification. Therefore, we turned to it in order to fill our knowledge base with information about flamenco album releases and recordings. Artists present in FlaBase were intended to be mapped with MusicBrainz artists. For every match, all content related to releases and recordings was gathered. After doing so, we obtained a total number of 814 releases and 9,942 recordings.

The information gathered from MusicBrainz is a little part of the actual flamenco discography. Therefore, to complement it we used a flamenco recordings database gathered by Rafael Infante and available at CICA website¹⁵ (Computing and Scientific Center of Andalusia). This database has information about releases from the early time of recordings until present time, counting 2,099 releases and 4,136 songs. For every song entry, a *cantaor* name is provided, and most of the times also guitarist and *palo*, which is a very valuable information to define flamenco recordings.

Finally, we supplied our knowledge base with information related to Andalusian towns and provinces. We gathered this information from the official database SIMA¹⁶ (Multi-territorial System of Information of Andalusia).

5.2 Entity Resolution

Entity Resolution (ER) is the problem of extracting, matching and resolving entity mentions in structured and unstructured data [10]. There are several approaches to tackle the ER problem. For the scope of this research, we selected a pair-wise classification approach based on string similarity between entity labels.

The first issue after gathering the data is to decide whether two entities from different sources are referring to the same one. Therefore, given two sets of entities A and B , the objective is to define an injective and non-surjective mapping function f between A and B that decides whether an entity $a \in A$ is the same as an entity $b \in B$. To do that, a string similarity metric $sim(a, b)$ based on the Ratcliff-Obershelp algorithm [16] has been defined. It measures

¹³ <http://andalucia.org>

¹⁴ <http://www.elartedevivirelflamenco.com/>

¹⁵ <http://flun.cica.es/index.php/grabaciones>

¹⁶ <http://www.juntadeandalucia.es/institutodeestadisticaycartografia/sima>

¹¹ <http://www.wikipedia.org>

¹² <http://es.dbpedia.org>

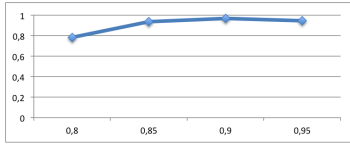


Figure 2. F-measure for different values of θ

the similarity between two entity labels and outputs a value between 0 and 1. We consider that a and b are the same entity if their similarity is bigger than a parameter θ . If there are two entities $b, c \in B$ that satisfy that $\text{sim}(a, b) \geq \theta$ and $\text{sim}(a, c) \geq \theta$, we consider only the mapping with the highest score. To determine the value of θ , we tested the method with several θ values over an annotated dataset of entity pairs. To create this dataset, the 291 artists gathered from Wikipedia were manually mapped to the 422 artists gathered from Andalusia.org, obtaining a total amount of 120 pair matches. As it is shown in Figure 2 the best F-measure (0,97) was obtained with $\theta = 0.9$. Finally, we applied the described method with $\theta = 0.9$ to all gathered entities from the three data sources. Thanks to the entity resolution process, we reduced the initial set of 1,462 artists and 132 *palos* to a set of 1,174 artists and 76 *palos*.

Once we had our artist entities resolved, we began to gather their related discographic information. First, we tried to find out the MusicBrainz ID of the gathered artists. Depending on the information about the entity, two different process were applied. First, every Wikipedia page, and its equivalent DBpedia resource, has a correspondent entity defined in Wikidata. Wikidata is a free linked database which acts as a structured data storage of Wikipedia. There are several properties in Wikidata that may link Wikidata items with MusicBrainz items. Thus, the equivalent Wikidata resource of a Wikipedia artist page may have a link to its corresponding MusicBrainz artist ID. Therefore, we looked for these relations and mapped all possible entities. For those artists without a direct link to MusicBrainz, we queried the MusicBrainz API by using the artist labels, and then applied our entity resolution method to the obtained results.

Finally, to integrate the discography database of CICA into our knowledge base, we applied the entity resolution method to the fields *cantaor*, guitarist and *palo* of each recording entry in the database. From the set of 202 *cantaores* and 157 guitarists names present in the recording entries, a total number of 78 *cantaores* and 44 guitarists were mapped to our knowledge base. The number of mapped artists was low due to differences between the way of labeling an artist. An artist name may be written using one or two surnames, or using a nickname. In the case of *palos*, there were 162 different *palos* in the database, 54 of which were mapped with the 76 of our knowledge base. These 54 *palos* correspond to an 80% of *palo* assignments present in the recording entries.

6. KNOWLEDGE EXTRACTION

Once the process of data acquisition is finished, the knowledge base is ready for use. However, there is an important amount of knowledge present in the data that has not been fully exploited. Texts gathered contain a huge epistemic potential that remains implicit. Consequently, to enhance the amount of structured data in FlaBase, a process of knowledge extraction has been carried out. This implicit knowledge may vary from biographical data, such as place and date of birth, to more complex semantic relations involving different entities. Three tasks play a key role in the process of knowledge extraction from non-structured text: named entity recognition (NER), named entity disambiguation (NED), and relation extraction (RE) [20]. In this research, we focus on the two first tasks. In what follows, a system for entity recognition and disambiguation is described and evaluated. Lastly, an information extraction process is applied to populate the knowledge base.

6.1 Named entity recognition and disambiguation

To extract implicit knowledge from a text, the first step is to semantically annotate it by identifying entity mentions. Named entity recognition is a task that seeks and classify words in text into pre-defined categories (e.g., person, organization, or place). Named entity disambiguation, also called entity linking, aims to determine what is actually a named entity present in a text. It generally does so by identifying it in a knowledge base of reference. NED can be addressed directly from the text, or applied to the output of a NER system. We propose a method that employs a combination of both approaches, depending on the category of the entity. For NER, we used the Stanford NER system [8], implemented in the library Stanford Core NLP¹⁷ and trained on Spanish texts. For NED we tried two different approaches. First, we looked for exact string matches between FlaBase entity labels and word n-grams extracted from the text. Second, we searched for exact string matches between FlaBase entity labels and the output of the NER system. In fact, we tried several combinations of both approaches until we obtained the most satisfactory one.

For the scope of this research, we focused on Spanish texts, as flamenco texts are mostly written in Spanish. Although there are many entity linking tools available, we decided to develop ours because state-of-the-art systems (e.g., Tag-me or Babelfy) are well-tuned for English texts, but do not perform well on Spanish texts, and even less with music texts of a specific domain. In addition, we wanted to have a system able to map entities to our knowledge base. Therefore, we developed a system able to detect and disambiguate three categories of entities: person, *palo* and location. Three different approaches were defined by combining NER and NED in different ways according to the category. First, directly applying NED to text. Second, disambiguating location and person entities from the

¹⁷<http://nlp.stanford.edu/software/corenlp.shtml>

Approach	Precision	Recall	F-measure
1) NED	0.829	0.694	0.756
2) NED + NER to PERS & LOC	0.739	0.347	0.472
3) NED + NER to LOC	0.892	0.674	0.767

Table 1. Precision, Recall and F-measure of NER+NED

NER output, and *palo* directly from text. Third, only disambiguating location entities from the NER output, and location and *palo* directly from text.

To determine which approach performs better, three artist biographies coming from three different data sources were manually annotated, having a total number of 49 annotated entities. We followed an evaluation methodology similar to the one used in KBP2014 Entity Linking Task¹⁸. Results on the different approaches are shown in Table 1. We observe that applying NER to entities of the person category before NED worsens performance significantly, as recall suddenly decrease by half. After manually analysing false negatives, we observed that this is caused because many artist names have definite articles between name and surname (e.g., *de*, *del*), and this is not recognized by the NER system. In addition, many artists have a nickname that is not interpreted as a person entity by the NER system. The best approach is the third (NED + NER to LOC), which is slightly better than the first (only NED) in terms of precision. This is due to the fact that many artists have a town name as a surname or as part of his nickname. Therefore, applying NED directly to text is misclassifying person entities as location entities. Thus, by adding a previous step of NER to location entities we have increased overall performance, as it can be seen on the F-measure values.

6.2 Knowledge base population

Biographical texts coming from different data sources have been stored in FlaBase. These texts are full of relevant information about FlaBase entities, but in an unstructured way. Thus, a process of information extraction is necessary to transform the unstructured information into structured knowledge. For the scope of this research, we focused on extracting two specific data: birth year and birth place, as they can be very relevant for anthropologic studies. We observed that this information is often in the very first sentences of the artist biographies, and always near the word *nació* (Spanish translation of “was born”). Therefore, to extract this information, we looked for this word in the first 250 characters of every biographical text. If it is found, we apply our entity linking method to this piece of text. If a location entity is found near the word “nació”, we assume that this entity is the place of birth of the biography subject. In addition, by using regular expressions, we look for the presence of a year expression in the neighborhood. If it is found, we assume it as the year of birth. If more than one year is found, we select the one with the smaller value.

To evaluate our approach, we tested the extraction of birth places in all texts coming from the web Andalucía.org (442 artists). We chose this subset because Andalucía.org

also provides specific information about artist origin that had been previously crawled and stored in FlaBase. However, we observed that in many occasions the artist origin provided by the data source was wrong. Therefore, we decided to manually annotate the province of precedence of these 442 artists for building ground truth data. After the application of the extraction process on the annotated test set, we obtained a precision value of 0,922 and a recall of 0,648. Therefore, we can state that our method is extracting biographic information with very high precision and quite reasonable recall. We finally applied the extraction process to all artist entities with biographical texts coming from any of the two flamenco crawled websites. Thus, from a total number of 1,123 artists coming from these data sources (95% of the artists in the knowledge base), 743 birth places and 879 birth years were extracted.

7. LOOKING AT THE DATA

7.1 Artist Relevance

We assume that an entity mention inside an artist biography means a semantic relation between the biography subject and the mentioned entity. Based on this assumption, we build a semantic graph by applying the following steps. First, each artist of the knowledge base is added to the graph as a node. Second, entity linking is applied to artist’s biographical texts. For every linked entity, a new node is created in the graph (only if it was not previously created). Next, an edge is added by connecting the artist entity node with the linked entity node. This way, a directed graph connecting the entities of FlaBase is finally obtained. Entities identified in a text can be seen as hyperlinks. Hence, algorithms to measure the relevance of nodes in a network of hyperlinks can be applied to our semantic graph [2]. In order to measure artist relevance, we applied PageRank [4] and HITS [11] algorithms to the obtained graph.

We built an ordered list with the top-10 entities of the different artist categories (*cantaor*, guitarist and *bailaor*) for the two algorithms. For evaluation purposes, we asked a flamenco expert to build a list of top-10 artists for each category according to his knowledge and the available bibliography. The concept of artist relevance is somehow subjective and there is no unified or consensual criteria for flamenco experts about who the most relevant artists are. Despite that, there is a high level of agreement among them on certain artists that should be on such a hypothetical list. Thus, the expert provided us with this list of hypothetical top-10 artists by category and we considered it as ground truth. We define precision as the number of identified artists in the resulting list that are also present in the ground truth list divided by the length of the list. We evaluated the output of the two algorithms by calculating precision over the entire list (top-10), and over the first five elements (top-5) (see Table 3). We observed that PageRank results (see Table 2) show the greatest agreement with the flamenco expert. High values of precision, specially for the top-5 list, indicates that the content gathered in FlaBase is

¹⁸ <http://nlp.cs.rpi.edu/kbp/2014/>

highly complete and accurate (see Table 3).

<i>Cantaor</i>	<i>Guitarist</i>	<i>Bailaor</i>
Antonio Mairena	Paco de Lucía	Antonio Ruiz Soler
Manolo Caracol	Ramón Montoya	Rosario
La Niña de los Peines	Niño Ricardo	Antonio Gades
Antonio Chacón	Manolo Santlúcar	Mario Maya
Camarón de la Isla	Sabicas	Carmen Amaya

Table 2. PageRank Top-5 artists by category

	Top-5	Top-10
PageRank	0.933	0.633
HITS Authority	0.6	0.4

Table 3. Precision values

7.2 Statistics

For the sake of completeness, some statistics on the data stored in FlaBase were calculated. Data shown in Figure 3 was produced out of the entity resolution process, while data shown in Figures 4 and 5 was calculated according to the populated data. In Figure 3 it is shown that the most representative *palos* are represented in the knowledge base, with a higher predominance of fandangos. We can observe in Figure 4 that most flamenco artists are from the Andalusian provinces of Seville and Cadiz. Finally, in Figure 5 we observe a higher number of artists in the data were born from the 30's to the 80's of the 20th century.

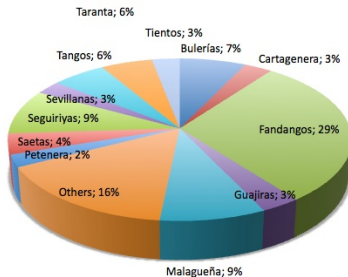


Figure 3. Songs by palo

8. CONCLUSIONS AND FUTURE WORK

A new knowledge base that contains information about flamenco music has been created and released. A process of automatic knowledge curation has been applied to combine information coming from different data sources. In addition, the knowledge base has been enriched with content extracted directly from texts by using a custom entity linking system. Using FlaBase data, artist relevance has been computed and compared to the flamenco experts' judgment. Precision values obtained reveals a high degree

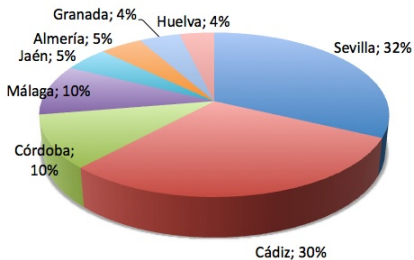


Figure 4. Artists by province of birth

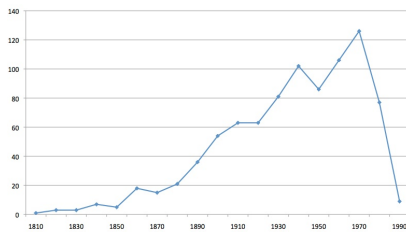


Figure 5. Artists by decade of birth

of coverage and a good quality of the knowledge base content.

There are still many avenues to be explored for future work. More websites can be exploited to increase coverage. The entity resolution step might be improved by increasing the amount of entity labels used, or by applying learning algorithms. A SPARQL endpoint might be created, letting users query FlaBase directly. In addition, implementing a collaborative environment for knowledge management would lead to an improvement in terms of completeness and data accuracy, as content might be added, checked and corrected directly by a community of users.

9. ACKNOWLEDGMENTS

This work was funded by the COFLA2 research project (Proyectos de Excelencia de la Junta de Andalucía, FEDER P12-TIC-1362). We thank Rafael Infante and José Ruiz Fuentes for the provided content.

10. REFERENCES

- [1] Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead, and Oren Etzioni. Open Information Extraction from the Web. In *International Joint Conferences on Artificial Intelligence*, pages 2670–2676, 2007.
- [2] Francesco Bellomi and Roberto Bonato. Network

- Analysis for Wikipedia. *Proceedings of Wikimania*, 2005.
- [3] Jose Blas Vega and Manuel Ríos Ruiz. *Diccionario enciclopédico ilustrado del flamenco*. Cinterco, Madrid, 1988.
- [4] Sergey Brin and Lawrence Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. *Computer Networks*, 30:107–117, 1998.
- [5] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. Toward an Architecture for Never-Ending Language Learning. In *AAAI*, 2010.
- [6] Anthony Fader, Luke Zettlemoyer, and Oren Etzioni. Open question answering over curated and extracted knowledge bases. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14*, pages 1156–1165, 2014.
- [7] Lola Fernández. *Teoría musical del flamenco*. Acordes Concert, Madrid, 2004.
- [8] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating Non-local Information into Information Extraction Systems by Gibbs Sampling. *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL 2005)*, pages 363–370, 2005.
- [9] J. M. Gamboa. *Una historia del flamenco*. Espasa-Calpe, Madrid, 2005.
- [10] Lise Getoor. Entity Resolution: Theory, Practice & Open Challenges. *Tutorial at AAI-12*, pages 2018–2019, 2012.
- [11] Jon M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46:604–632, 1999.
- [12] J.L. Navarro and M. Roperro. *Historia del flamenco*. Ed. Tartessos, Sevilla, 1995.
- [13] Sergio Oramas. Harvesting and Structuring Social Data in Music Information Retrieval. *Extended Semantic Web Conference (ESWC). Lecture Notes in Computer Science*, 8465:817–826, 2014.
- [14] M Cristina Pattuelli, Matt Miller, Leanora Lange, Sean Fitzell, and Carolyn Li-Madeo. Crafting Linked Open Data for Cultural Heritage: Mapping and Curation Tools for the Linked Jazz Project. *Code4Lib Journal*, page 4, 2013.
- [15] Alastair Porter, Mohamed Sordo, and Xavier Serra. Dunya: A System for Browsing Audio Music Collections Exploiting Cultural Context. *14th International Society for Music Information Retrieval Conference (ISMIR 2013)*, 2013.
- [16] John W Ratcliff and David Metzner. Pattern matching: The gestalt approach. *Dr. Dobb's Journal*, 13:46–72, 1988.
- [17] Markus Schedl, Emilia Gómez, and Julián Urbano. Music Information Retrieval: Recent Developments and Applications. *Foundations and Trends® in Information Retrieval*, 8(2-3):127–261, 2014.
- [18] Xavier Serra. Data gathering for a culture specific approach in MIR. *Proceedings of the 21st international conference companion on World Wide Web - WWW '12 Companion*, page 867, 2012.
- [19] Xavier Serra. Creating Research Corpora for the Computational Study of Music : the case of the CompMusic Project. *53rd International Conference: Semantic Audio (January 2014)*, pages 1–9, 2014.
- [20] Ricardo Usbeck, Axel-cyrille Ngonga Ngomo, R Michael, Daniel Gerber, Sandro Athaide Coelho, and Andreas Both. AGDISTIS - Graph-Based Disambiguation of Named Entities using Linked Data. *The Semantic Web – ISWC 2014*, 2014.

DISCOVERY OF SYLLABIC PERCUSSION PATTERNS IN TABLA SOLO RECORDINGS

Swapnil Gupta*

swapnil.gupta01@student.upf.edu

Ajay Srinivasamurthy*

ajays.murthy@upf.edu

Manoj Kumar†

manojpamk@gmail.com

Hema A. Murthy†

hema@cse.iitm.ac.in

Xavier Serra*

xavier.serra@upf.edu

*Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain

†DONlab, Indian Institute of Technology Madras, Chennai, India

ABSTRACT

We address the unexplored problem of percussion pattern discovery in Indian art music. Percussion in Indian art music uses onomatopoeic oral mnemonic syllables for the transmission of repertoire and technique. This is utilized for the task of percussion pattern discovery from audio recordings. From a parallel corpus of audio and expert curated scores for 38 tabla solo recordings, we use the scores to build a set of most frequent syllabic patterns of different lengths. From this set, we manually select a subset of musically representative query patterns. To discover these query patterns in an audio recording, we use syllable-level hidden Markov models (HMM) to automatically transcribe the recording into a syllable sequence, in which we search for the query pattern instances using a Rough Longest Common Subsequence (RLCS) approach. We show that the use of RLCS makes the approach robust to errors in automatic transcription, significantly improving the pattern recall rate and F-measure. We further propose possible enhancements to improve the results.

1. INTRODUCTION

In many music cultures, music is sometimes transmitted partly through speech, using what are variously called vocables, oral mnemonics, solfège, etc [12]. In the case of several percussion traditions, the choice of vowels and consonants is such that the syllables closely represent the underlying acoustic phenomenon they represent. The term *acoustic-iconic mnemonic* systems coined by Hughes [12] explains this mnemonic based syllable systems where the core aspect is the similarity of the phonetic features of the syllables with the acoustic properties of the sounds they represent. A well studied example of such a system is the tabla, where the repertoire and technique is transmitted with the help of a system based on onomatopoeic oral syllables [18]. In this paper, we explore the use of the mnemonic syllable system of tabla for the discovery of percussion patterns. The use of these mnemonics allows us to work with a

musically relevant representation that truly reflects the underlying timbre, articulation and dynamics of the patterns played.

Automatic discovery of patterns is a relevant Music Information Retrieval (MIR) task. It has applications in enriched and informed music listening, enhanced appreciation for listeners, in music training, and in aiding musicologists working on such music cultures. We use the onomatopoeic oral mnemonic syllables to represent, transcribe and search for patterns in audio recordings of tabla solos. We first build a set of query patterns from the corpus of scores in our dataset. Given an audio recording, we automatically transcribe it into a sequence of syllables. We then propose a method for searching the query patterns in the automatically transcribed score using approximate string search. We also propose several extensions to improve the search performance. We first provide a brief introduction to tabla.

1.1 Tabla and its solo performances

Tabla is the main rhythm accompanying instrument in Hindustani music, the art music tradition from North India. It consists of two drums: a left hand bass drum called the *bāyān* or *diggā* and a right hand drum called the *dāyān* that can produce a variety of pitched sounds [15]. To showcase the nuances of the *tāl* (the rhythmic framework of Hindustani music) as well as the skill of the percussionist with the tabla, Hindustani music performances feature tabla solos. A tabla solo is intricate and elaborate, with a variety of pre-composed forms used for developing further elaborations. There are specific principles that govern these elaborations [10, p. 42]. Musical forms of tabla such as the *thēkā*, *kāyādā*, *palatā*, *rēlā*, *pēškār* and *gaṭ* are a part of the solo performance and have different functional and aesthetic roles in a solo performance.

Playing a tabla is taught and learned through the use of onomatopoeic oral mnemonic syllables called the *bōl*, which are vocal syllables corresponding to different timbres that can be produced on the tabla. However, several *bōls* correspond to the same stroke played on the tabla, creating a many *bōl* to same timbre mapping, which can be exploited to discover acoustically similar patterns. Though the primary function of the *bōls* is to provide a representation system, a rhythmic vocal recitation of the *bōls*, which requires high skills, is inserted into solo performances for music appreciation.

Tabla has different stylistic schools called *gharanās*. The



© Swapnil Gupta, Ajay Srinivasamurthy, Manoj Kumar, Hema A. Murthy, Xavier Serra.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Swapnil Gupta, Ajay Srinivasamurthy, Manoj Kumar, Hema A. Murthy, Xavier Serra. "Discovery of Syllabic Percussion Patterns in Tabla Solo Recordings", 16th International Society for Music Information Retrieval Conference, 2015.

Sym.	bōls	Sym.	bōls
DA	D, DA, DAA	NA	N, NA, TAA, TU
KI	KA, KAT, KE, KI, KII	DIN	DI, DIN, DING, KAR, GHEN
GE	GA, GHE, GE, GHI, GI	KDA	KDA, KRA, KRI, KRU
TA	TA, TI, RA	TIT	CHAP, TIT

Table 1: The bōls used in tabla, their grouping, and the symbol we use for the syllable group in this paper. The symbols DHA, DHE, DHET, DHI, DHIN, RE, TE, TII, TIN, TRA have a one to one mapping with a syllable of the same name and hence not shown in the table.

repertoires of major gharānās of tabla differ in aspects such as the use of specific bōls, the dynamics of strokes, ornamentation and rhythmical phrases [4, p. 60]. But there are also many similarities due to the fact that the same forms and standard phrases reappear across these repertoires [10, p. 52]. This enables in creation of a library of standard phrases or patterns across compositions of different gharānās.

1.2 Previous Work

Early research related to tabla focused mainly on stroke transcription, as seen in the work of Gillet [9]. Chordia [6] extended the work adding additional features and classifiers, using a larger and more diverse dataset. The use of tabla syllables in a predictive model for tabla stroke sequence was also demonstrated recently by Chordia et al. [7]. Recent work in transcription has been reported for Mridangam, the percussion accompaniment used in South Indian Carnatic music, by Kuriakose et al. [13] and Anantapadmanabhan et al. [1]. The transcription task has a definite analogy to speech recognition and we can apply several tools and knowledge from this well explored research area with many state of the art algorithms and systems [11].

There is significant literature on pattern search and retrieval from percussion solos. Nakano et al. [16] address the problem of drum pattern retrieval using an HMM based approach using onomatopoeia as the representation for drum patterns, retrieving known fixed sequences from a library of drum patterns with snare and bass drums. We use a similar approach, the main difference being that we use a musically well grounded syllabic representation. Recently, Srinivasamurthy et al. [20] demonstrated the use of syllable level HMM followed by a string edit distance to transcribe and classify percussion patterns in Beijing Opera. Tsunoo et al. [21] also demonstrated a music classification task using K-means clustering of bar-long percussive patterns and bass lines extracted using one-pass dynamic programming. While the last two mentioned approaches aim at classification of patterns, we address the general task of retrieving patterns from recordings of full length solo compositions.

Transcription is often inaccurate with many errors, and any pattern search on transcribed data needs to use approximate string search algorithms. There are several attempts to deal with search in symbolic sequences [22]. Well explored techniques such as longest common subsequence (LCS) do not consider the local correlation while searching for a sub-

sequence [14]. To overcome this limitation, Lin et al. [14] proposed a novel Rough Longest Common Subsequence (RLCS) method for music matching. Dutta et al. [8] used a modified version of RLCS for motif spotting in ālāpanas of Carnatic music. We propose to use a similar approach with minor modifications to suit the symbolic domain specific to our use case. To the best of our knowledge, this is the first work to explore syllabic pattern discovery as applied to tabla solos in Hindustani music.

2. PROBLEM FORMULATION

We formulate the problem of discovery of percussion patterns in tabla solo recordings. We present a general framework for the task, while outlining some of the challenges. The approach we explore in this paper is to use syllables to define, transcribe, and eventually search for percussion patterns. We build a fixed set of syllabic query patterns. Given an audio recording, we obtain a time-aligned syllabic transcription using syllable level timbral models. For each of the query patterns in the set, we then perform an approximate search on the output transcription to obtain the locations of the patterns in the audio recording. We describe each of the steps in detail.

We first compile a comprehensive set of syllables in tabla. Although, the syllables vary marginally within and across gharānās, several bōls can represent the same stroke on the tabla. To address this issue, we grouped the full set of 41 syllables into timbrally similar groups resulting into a reduced set of 18 syllable groups as shown in Table 1. Though each syllable on its own has a functional role, this timbral grouping is presumed to be sufficient for discovery of percussion patterns. For the remainder of the paper, we limit ourselves to the reduced set of syllable groups and use them to represent patterns. For convenience, when it is clear from the context, we call the syllable groups as just syllables and denote them by the symbols in Table 1. Further, we use bōls and syllables interchangeably. Let the set of syllables be denoted as $\mathcal{S} = \{S_1, S_2, \dots, S_M\}$, $M = 18$.

A percussion pattern is not well defined and varied definitions can exist. Here, we use a simplistic definition of a pattern, as a sequence of syllables. A pattern is defined as $P_k = [s_1, s_2, \dots, s_{L_k}]$ where $s_k \in \mathcal{S}$ and L_k is the length of P_k . Though, for defining patterns, it is important to consider the relative and absolute durations of the constituent syllables, as well as the metrical position of the pattern in the tāl, we use a simple definition and leave a more comprehensive definition for future work. In this paper, we take a data driven approach to build a set of K query patterns, $\mathcal{P} = \{P_1, P_2, \dots, P_K\}$.

Given an audio recording $x[n]$, it is first transcribed into a sequence of time-aligned syllables, $T_x = [(t_1, s_1), (t_2, s_2), \dots, (t_{L_x}, s_{L_x})]$, where t_i is the onset time of syllable s_i . The task of syllabic transcription has a significant analogy to connected word speech recognition using word models. Syllables are analogous to words and a percussion pattern to a sentence - a sequence of words. Finally, given a query pattern P_k of length L_k , we search for the pattern in the output syllabic transcription T_x , to retrieve the subsequences $p_k^{(n)}$ in T_x ($n = 1, \dots, N_k$) that match the query, where

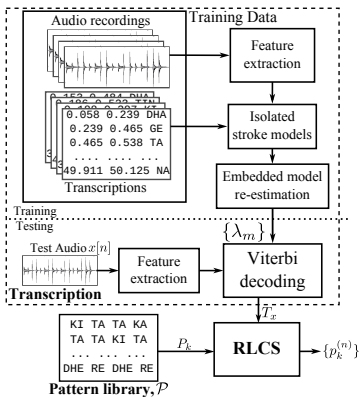


Figure 1: The block diagram of the approach

N_k is the number of retrieved matches for P_k . We use $p_k^{(n)}$ and the corresponding onset times from T_x to extract audio segments corresponding to the retrieved syllabic patterns. Syllabic transcription is often not exact and it can have common transcription errors such as insertions, substitutions and deletions, to handle which we need an approximate search algorithm.

3. DATASET

To evaluate our approach to percussion pattern discovery, we need a parallel corpus with time-aligned scores and audio recordings. These are useful both for building isolated stroke timbre models and for a comprehensive evaluation of the approach. We built a dataset comprising audio recordings, scores and time aligned syllabic transcriptions of 38 tabla solo compositions of different forms in *fiñtal* (a metrical cycle of 16 time units). The compositions were obtained from the instructional video DVD *Shades Of Tabla* by Pandit Arvind Mulgaonkar¹. Out of the 120 compositions in the DVD, we chose 38 representative compositions spanning all the gharānās of tabla (Ajrada, Benaras, Dilli, Lucknow, Punjab, Farukhabad). The booklet accompanying the DVD provides a syllabic transcription for each composition. We used Tesseract [19], an open source Optical Character Recognizer (OCR) engine to convert printed scores to a machine readable format. The scores obtained from OCR were manually verified and corrected for errors, adding the the *vibhāgs* (sections) of the *tāl* to the syllabic transcription. The score for each composition has additional metadata describing the gharānā, composer and its musical form.

We extracted audio from the DVD video and segmented the audio for each composition from the full audio recording. The audio recordings are stereo, sampled at 44.1 kHz and have a soft harmonium accompaniment. A time aligned syllabic transcription for each score and audio file pair was obtained using a spectral flux based onset detector [3] fol-

ID	Pattern	L	Count
1	DHE, RE, DHE, RE, KI, TA, TA, KI, NA, TA, TA, KI, TA, TA, KI, NA	16	47
2	TA, TA, KI, TA, TA, KI, TA, TA, KI, TA, TA, KI, TA, TA, KI, TA	16	10
3	TA, KI, TA, TA, KI, TA, TA, KI	8	61
4	TA, TA, KI, TA, TA, KI	6	214
5	TA, TA, KI, TA	4	379
6	KI, TA, TA, KI	4	450
7	TA, TA, KI, NA	4	167
8	DHA, GE, TA, TA	4	97

Table 2: Query Patterns, their ID (k), length (L) and the number of instances in the dataset (Total instances: 1425)

lowed by manual correction by the authors. The dataset contains about 17 minutes of audio with over 8200 syllables. The dataset is freely available for research purposes through a central online repository².

4. APPROACH

The block diagram in Figure 1 shows us the overall approach. It comprises three major steps: *building a set of query patterns, transcription, and search*. In the following sections, we describe each of these in detail.

4.1 Building a set of query patterns

A data driven approach is taken to create a set of query patterns of length $L = 4, 6, 8, 16$. These lengths were chosen based on the structure of *fiñtal* for different *layas* (tempo classes) [4, p. 126]. Using the simple definition of a pattern as a sequence of syllables, we use the scores of the compositions to generate all the L length patterns that occur in the score collection. We sort them by their frequency of occurrence to get an ordered set of patterns for each stated length. We then manually choose musically representative patterns from this ordered set of most commonly occurring patterns to form a set of query patterns. Table 2 shows the chosen patterns, their length and their count in the dataset, leading to a total of 1425 instances. We want a diverse collection of patterns to test if the algorithms generalize. Hence we choose patterns that have a varied set of syllables that have different timbral characteristics, like syllables that are harmonic (DHA), syllables played with a flam (DHE, RE) and syllables having bass (GE).

4.2 Transcription

Some *bōls* of tabla may be pronounced with a different vowel or consonant depending on the context, without altering the drum stroke [5]. Furthermore, the *bōls* and the strokes vary across different gharānās, making the task of transcription of tabla solos challenging. To model the timbral dynamics of syllables, we build an HMM for each syllable (analogous to a word-HMM). We use these HMMs along with a language model to transcribe an input audio solo recording into a sequence of syllables.

¹ <http://musicbrainz.org/release/220c5efc-2350-43dd-95c6-4870dc6851f5>

² <http://commusic.upf.edu/tabla-solo-dataset>

The stereo audio is converted to mono, since there is no additional information in stereo channels. We use the MFCC features to model the timbre of the syllables. To capture the temporal dynamics of syllables, we add the velocity and the acceleration coefficients of the MFCC. The 13 dimensional MFCC features (including the 0th coefficient) are computed from the audio with a frame size of 23.2 ms and a shift of 5.8 ms. We also explore the use of energy (as measured by the 0th MFCC coefficient) in transcription performance. Hence we have two sets of features, MFCC_0_D_A, the 39 dimensional feature including the 0th, delta and double-delta coefficients, and MFCC_D_A, the 36 dimensional vector without the 0th coefficient.

Using the features extracted from training audio recordings, we model each syllable S_u using a 7-state left-to-right HMM $\{\lambda_u\}$, $1 \leq u \leq U (= 18)$, including an entry and an exit non-emitting states. The emission density of each emitting state is modeled with a three component Gaussian Mixture Model (GMM) to capture the timbral variability in syllables. We experimented with higher number of components in the GMMs, but with little performance improvement. We use the time aligned syllabic transcriptions and the audio recordings in the parallel corpus to do an isolated HMM training for each syllable. We then use these HMMs further in an embedded model Baum-Welch re-estimation to get the final syllable HMMs.

Tabla solos are built hierarchically using short phrases, and hence some bōls tend to follow a bōl more often than others. In such a scenario, a language model can improve transcription. In addition to a flat language model with uniform unigram and transition probabilities, i.e. $p(s_1 = S_u) = 1/U$ and $p(s_{i+1} = S_v | s_i = S_u) = 1/U$, with $1 \leq u, v \leq U$ and i being the sequence index, we explore the use of a bigram language model learned from data.

For testing, we treat the feature sequence extracted from test audio file to have been generated from a first order time-homogeneous discrete Markov chain, which can consist of any finite length sequence of syllables. From the extracted feature sequence, we use the HMMs $\{\lambda_u\}$ and a syllable network constructed from the language model to do a Viterbi (forced) alignment, which aims to provide the best sequence of syllables and their onsets T_x . All the transcription experiments were done using the HMM Toolkit (HTK) [23].

4.3 Pattern Search

The automatically transcribed output syllable sequence T_x is used to search for the query patterns. Transcription is often inaccurate in both the sequence of syllables and in the exact onset times of the transcribed syllables. We need to handle both these errors in a pattern search task from audio. We primarily focus on the errors in syllabic transcription in this paper. We use the syllable boundaries output by the Viterbi algorithm, without any additional post processing. We can improve the output syllable boundaries using an onset detector [3], but we leave this task to future work.

There are three main kinds of errors in the automatically transcribed syllable sequence: Insertions (I), Deletions (D), and Substitutions (B). Further, the query pattern is to be searched in the whole transcribed composition, where sev-

eral instances of the query can occur. Rough Longest Common Subsequence (RLCS) method is a suitable choice for such a case. RLCS is a subsequence search method that searches for roughly matched subsequences while retaining the local similarity [14]. We make further enhancements to RLCS to handle the I, D and B errors in transcription.

We use a modified version of the RLCS approach as proposed by Lin et al. [14] with changes proposed by Dutta et al. [8] to handle substitution errors. We propose a further enhancement to handle insertions and deletions, and explore its use in the current task. We first present a general form of RLCS and then discuss different variants of the algorithm.

Given a query pattern P_k of length L_k and a reference sequence (transcribed syllable sequence) T_x of length L_x , RLCS uses a dynamic programming approach to compute a score matrix (of size $L_x \times L_k$) between the reference and the query with a rough length of match. We can use a threshold on the score matrix to obtain the instances of the query occurring in the reference. We can then use the syllable boundaries in the output transcription and retrieve the audio segment corresponding to the match.

For the ease of notation, we index the transcribed syllable sequence T_x with i and the query syllable sequence P_k with j . We compute the rough and actual length of the subsequence matches similar to the way computed by Dutta et al. [8]. At every position (i, j) , a syllable is included into the matched subsequence if $d(s_i, s_j) < \delta$, where $d(s_i, s_j)$ is the timbral distance between the syllables at positions i and j in the transcription and query, respectively. δ is the threshold distance below which the two syllables are said to be equivalent. The matrices of rough length of match (C) and the actual length of match (C^a) are updated as,

$$C(i, j) = C(i-1, j-1) + (1 - d(s_i, s_j)) \cdot \mathbb{1}_d \quad (1)$$

$$C^a(i, j) = C^a(i-1, j-1) + \mathbb{1}_d \quad (2)$$

where, $\mathbb{1}_d$ is an indicator function that takes a value of 1 if $d(s_i, s_j) < \delta$, else 0. The matrix C thus contains the length of rough matches ending at all combinations of the syllable positions in reference and the query. The rough length and an appropriate distance measure handles the substitution errors during transcription. To penalize insertion and deletion errors, we compute a “density” of match using two measures called the Width Across Reference (WAR) and Width Across Query (WAQ), respectively. The WAR (R) and WAQ (Q) matrices are initialized to $R_{i,j} = Q_{i,j} = 0$ when $i, j = 0$, and propagated as,

$$R_{i,j} = \begin{cases} R_{i-1,j-1} + 1 & d(s_i, s_j) < \delta \\ R_{i-1,j} + 1 & d(s_i, s_j) \geq \delta, C_{i-1,j} \geq C_{i,j-1} \\ R_{i,j-1} & d(s_i, s_j) \geq \delta, C_{i-1,j} < C_{i,j-1} \end{cases} \quad (3)$$

$$Q_{i,j} = \begin{cases} Q_{i-1,j-1} + 1 & d(s_i, s_j) < \delta \\ Q_{i-1,j} & d(s_i, s_j) \geq \delta, C_{i-1,j} \geq C_{i,j-1} \\ Q_{i,j-1} + 1 & d(s_i, s_j) \geq \delta, C_{i-1,j} < C_{i,j-1} \end{cases} \quad (4)$$

Here, $R_{i,j}$ is the length of substring containing the subsequence match ending at the i^{th} and the j^{th} position of the reference and the query, respectively. $Q_{i,j}$ represents a simi-

lar measure in the query. When incremented, $\mathbf{R}_{i,j}$ and $\mathbf{Q}_{i,j}$ are incremented by 1 similar to the way formulated by Lin et al. [14]. At the same time, the increment is done based on the conditions formulated by Dutta et al. [8].

Using the rough length of match (\mathbf{C}), actual length of match (\mathbf{C}^*), and width measures (\mathbf{R} and \mathbf{Q}), we compute a score matrix σ that incorporates penalties for substitutions, insertions, deletions, and additionally, the fraction of the query matched.

$$\sigma_{i,j} = \begin{cases} \left[\beta \cdot f\left(\frac{C_{i,j}}{R_{i,j}}\right) + (1-\beta) \cdot f\left(\frac{C_{i,j}}{Q_{i,j}}\right) \right] \frac{C_{i,j}^*}{L_k} & \text{if } \frac{C_{i,j}^*}{L_k} > \rho \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

where $\sigma_{i,j}$ is the score for the match ending at the i^{th} and the j^{th} position of the reference and the query, respectively. f is a warping function for the rough match length densities $\mathbf{R}_{i,j}$ in the reference and $\mathbf{Q}_{i,j}$ in the query. The parameter β controls their weights in the convex combination for score computation. The term $\frac{C_{i,j}^*}{L_k}$ is the fraction of the query length matched and is used for thresholding the minimum fraction of the query to be matched.

Starting with all combinations of i and j as the end points of the match in the reference and the query, respectively, we perform a traceback to get the starting points of the match. RLCS algorithm outputs a match when the score is more than a score threshold ψ . However, with a simple score thresholding, we get multiple overlapping matches, from which we select the match with the highest score. If the scores of multiple overlapping matches are equal, we select the ones that have the lowest width (WAR). This way, we obtain a match that has the highest score density. We use these non-overlapping matches and the corresponding syllable boundaries to retrieve the audio patterns.

4.3.1 Variants of RLCS

The generalized RLCS provides a framework for subsequence search. The parameters ρ , β , ψ and δ can be tuned to make the algorithm more sensitive to different kinds of transcription errors. The variants we consider here use different distance measures $d(s_i, s_j)$ in Eqn (1) to handle substitutions and different functions $f(\cdot)$ in Eqn (5) to handle insertions and deletions. We explore these variants for the current task and evaluate their performance.

In a default RLCS configuration (RLCS₀), we only consider exact syllable matches. We set $\delta = 1$ and use a binary distance metric based on the syllable label, i.e. $d(s_i, s_j) = 0$ if $s_i = s_j$, and 1 otherwise. Further, an identity warping function, $f(y) = y$ is used.

The rough length match densities can be transformed using a non-linear warping function to penalize low density values more than the higher ones, leading to another variant of RLCS (RLCS _{κ}). In this paper, we only explore warping functions of the form,

$$f(y) = \frac{e^{\kappa y} - 1}{e^{\kappa} - 1} \quad (6)$$

where $\kappa > 0$ is a parameter to control warping, larger values of κ lead to more deviation from an identity transformation. RLCS₀ is a limiting case of RLCS _{κ} when $\kappa \rightarrow 0$.

We hypothesize that the substitution errors in transcription are due to the confusion between timbrally similar syllables. A timbral similarity (distance) measure between the syllables can thus be used to make an RLCS algorithm robust to specific kinds of substitution errors. In essence, we want to disregard and give a greater allowance for substitutions between timbrally similar syllables during RLCS matching. Computing timbral similarity is a wide area of research and has many different proposed methods [17], but we restrict ourselves to a basic timbral distance measure: the Mahalanobis distance between the cluster centers obtained using a K-means clustering of MFCC features (with 3 clusters) from isolated audio examples of each syllable [2]. We call this variant of RLCS as RLCS _{δ} and experiment with different thresholds δ . For better reproducibility of the work in this paper, an implementation of the different variants of RLCS described is available³.

5. EXPERIMENTS AND RESULTS

We experiment with different sets of features and language models for transcription. With the best performing transcription configuration, we experiment with different RLCS variants and report their performance. We first describe the evaluation measures used in this paper.

5.1 Evaluation measures

We use the ground truth time aligned syllabic transcriptions to evaluate both the transcription and pattern search algorithms. We evaluate transcription performance using the measures often used in speech recognition, Correctness (Corr.) and Accuracy (Accu.). Given the ground truth transcription T_g^* of length N , the transcribed sequence T_x , and the number of insertions, deletions and substitutions as N_I , N_D , and N_B , respectively, we compute Corr. = $(N - N_D - N_B) / N$ and Accu. = $(N - N_D - N_B - N_I) / N$. The Correctness measure penalizes deletions and substitutions, while Accuracy measure additionally penalizes insertions.

For pattern retrieval, we don't evaluate the accuracy of boundary segmentation. However, we call a retrieved pattern from RLCS as *correctly retrieved* if it has at least a 70% overlap with the pattern instance in ground truth. To evaluate pattern search performance, we use the standard information retrieval measures precision (the ratio between the number of correctly retrieved patterns and all retrieved patterns) and recall (the ratio between number of correctly retrieved patterns and the patterns in the ground truth). The harmonic mean of precision and recall, called the F-measure is also reported.

5.2 Results and Discussion

The transcription results shown in Table 3 are the mean values in a leave-one-out cross validation over the dataset. We experimented with the two different MFCC features (MFCC_D_A and MFCC_0_D_A) and two language models (a flat model and a bigram learnt from data). Overall, we see a best Accuracy of 53.13%, which justifies the use of a robust approximate string search algorithm for pattern retrieval. The use of a bigram language model learned from data improves the transcription performance. We see that

³ <http://compmusic.upf.edu/ismir-2015-tabla>

	Feature	Corr.	Accu.
Flat language model	MFCC_D_A	64.07	45.01
	MFCC_0_D_A	64.26	49.27
Bigram language model	MFCC_D_A	65.53	49.97
	MFCC_0_D_A	66.23	53.13

Table 3: Transcription results showing the Correctness (Corr.) and Accuracy (Accu.) measures (in percentage) for different features and language models. In each column, the values in bold are statistically equivalent to the best result (in a paired-sample t-test at 5% significance levels).

the Accuracy measure is lower than the Correctness measure, which shows that there are a significant number of insertion errors in transcription. We use the output transcriptions from the best performing combination (MFCC_0_D_A and a bigram language model) to report the performance of the RLCS variants.

To form a baseline for string search performance with the output transcriptions, we used an exact string search algorithm and report its performance in Table 4 (shown as Baseline). We see that the baseline has a precision that is similar to transcription performance, but a very poor recall leading to a poor F-measure.

To establish the optimum parameter settings for RLCS, we performed a grid search over the values of β , ρ and ψ with RLCS₀. β and ψ are varied in the range 0 to 1. To ensure that the minimum length of the pattern matched is at least 2, we varied ρ between $1.1 \min(L_k)$ and 1.

β is the convex sum parameter for the contribution of the rough match length density of the reference and the query towards the final score. With increasing β , we give more weight to the reference length ratio, allowing more insertions. We observed a poor true positive rate with larger β , and hence we validate the observation that insertion errors contribute to a majority of transcription errors.

The best average F-measure over all the query patterns in an experiment using RLCS₀ is reported in Table 4. We see that RLCS₀ improves the recall, but with a lower precision and an improved F-measure, showing that the flexibility in approximate matching provided by RLCS comes at the cost of additional false positives. The values of ρ , β and ψ that give the best F-measure are then fixed for all subsequent experiments to compare the performance of the proposed RLCS variants.

It is observed that the patterns composed of smaller repetitive patterns (and hence having ambiguous boundaries) result in a poor precision (e.g. P_2 and P_3 in Table 2 with a precision of 0.108 and 0.239, respectively). P_1 in Table 2, on the contrary, has non-ambiguous boundaries leading to a good precision of 0.692. The effect of the length of a pattern on precision is also evident. Small patterns (with $L = 4$) that have non-ambiguous boundaries (e.g. P_8 in Table 2 with a precision of 0.384) have a poor precision as compared to longer patterns with non-ambiguous boundaries (e.g. P_1 in Table 2). The reason for this is that the smaller patterns are more prone to errors as the search algorithm has to match a lower number of syllables.

The results with other variants of RLCS are also reported in Table 4. The results from RLCS _{δ} show that the use of

Variant	Parameter	Precision	Recall	F-measure
Baseline	-	0.479	0.254	0.332
RLCS ₀	$\delta = 1$	0.384	0.395	0.389
RLCS _{δ}	$\delta = 0.3$	0.139	0.466	0.214
RLCS _{δ}	$\delta = 0.6$	0.0837	0.558	0.145
RLCS _{κ}	$\kappa = 1$	0.412	0.350	0.378
RLCS _{κ}	$\kappa = 4$	0.473	0.268	0.342
RLCS _{κ}	$\kappa = 7$	0.482	0.259	0.336
RLCS _{κ}	$\kappa = 9$	0.481	0.258	0.335

Table 4: Performance of different RLCS variants using the best performing parameter settings for RLCS₀ ($\rho = 0.875$, $\beta = 0.76$ and $\psi = 0.6$).

a timbral syllable distance measure with higher threshold δ further improves the recall, but with a much lower precision and F-measure. Although we find matches that have substitution errors using the distance measure, we retrieve additional matches that do not have substitution errors contributing to additional false positives. On the contrary, using a non-linear warping function $f(\cdot)$ in RLCS _{κ} improves the precision with a higher value of κ . The penalties on matches with higher number of insertions and deletions is high and they are left out, leading to good precision at the cost of recall. We observe that both the above mentioned variants improve either precision or recall at the cost of the other measure. They need further exploration with better timbral similarity measures to be combined in an effective way to improve the search performance.

6. SUMMARY

We addressed the unexplored problem of a discovering syllabic percussion patterns in Tabla solo recordings. The presented formulation used a parallel corpus of audio recordings and syllabic scores to create a set of query patterns, that were searched in an automatically transcribed (into syllables) piece of audio. We used a simplistic definition of a pattern and explored RLCS based subsequence search algorithm, using an HMM based automatic transcription. Compared to a baseline, we showed that the use of approximate string search algorithms improved the recall at the cost of precision. Additionally, proposed variants improved either the precision or recall, but do not provide a significant improvement in the F-measure over the basic RLCS.

For future work, we aim to improve syllable boundaries output by transcription using onset detection. Inclusion of the rhythmic information can be an interesting aspect in defining and discovering percussion patterns, and will help in comprehensively evaluating the task of pattern discovery. The next steps would be to incorporate better timbral similarity measures and inclusion of segment boundaries into the RLCS algorithm that effectively combines the proposed variants.

Acknowledgments

This work is partly supported by the European Research Council under the European Union's Seventh Framework Program, as a part of the CompMusic project (ERC grant agreement 267583). The authors thank Pandit Arvind Mulgaonkar for sharing the DVD of Tabla solo recordings.

7. REFERENCES

- [1] A. Anantapadmanabhan, A. Bellur, and H. A. Murthy. Modal analysis and transcription of strokes of the mridangam using non-negative matrix factorization. In *Proc. of the 38th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 181–185, Vancouver, Canada, May 2013.
- [2] J. J. Aucouturier and F. Pachet. Music similarity measures: What’s the use? In *Proc. of 3rd International Conference on Music Information Retrieval*, pages 157–163, Paris, France, 2002.
- [3] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and Mark B. Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047, Sept 2005.
- [4] S. Beronja. *The Art of the Indian tabla*. Rupa and Co. New Delhi, 2008.
- [5] A. Chandola. *Music as Speech: An Ethnomusicological Study of India*. Navrang, 1988.
- [6] P. Chordia. Segmentation and recognition of tabla strokes. In *Proc. of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 107–114, London, UK, September 2005.
- [7] P. Chordia, A. Sastry, and S. Şentürk. Predictive tabla modelling using variable-length markov and hidden markov models. *Journal of New Music Research*, 40(2):105–118, 2011.
- [8] S. Dutta and H. A. Murthy. A modified rough longest common subsequence algorithm for motif spotting in an alapana of carnatic music. In *Proc. of the 20th National Conference on Communications (NCC)*, pages 1–6, Kanpur, India, February 2014.
- [9] O. Gillet and G. Richard. Automatic labelling of tabla signals. In *Proc. of the 4th International Conference on Music Information Retrieval (ISMIR)*, Baltimore, USA, October 2003.
- [10] R. S. Gottlieb. *Solo Tabla Drumming of North India: Its Repertoire, Styles, and Performance Practices*. Motilal Banarsidass Publishers, 1993.
- [11] X. Huang and L. Deng. An overview of modern speech recognition. In N. Indurkha and F. J. Damerou, editors, *Handbook of Natural Language Processing*, Chapman & Hall/CRC Machine Learning & Pattern Recognition, pages 339–366. Chapman and Hall/CRC, 2nd edition, February 2010.
- [12] D. Hughes. No nonsense: the logic and power of acoustic-iconic mnemonic systems. *British Journal of Ethnomusicology*, 9(2):93–120, 2000.
- [13] J. Kuriakose, J. C. Kumar, P. Sarala, H. A. Murthy, and U. K. Sivaraman. Akshara transcription of mridangam strokes in carnatic music. In *Proc. of the 21st National Conference on Communication (NCC)*, Mumbai, India, February 2015.
- [14] H. Lin, H. Wu, and C. Wang. Music matching based on rough longest common subsequence. *Journal Information Science and Engineering*, 27(1):95–110, 2011.
- [15] M. Miron. Automatic Detection of Hindustani Talas. Master’s thesis, Music Technology Group, Universitat Pompeu Fabra, 2011.
- [16] T. Nakano, J. Ogata, M. Goto, and Y. Hiraga. A drum pattern retrieval method by voice percussion. In *Proc. of the 5th International Conference on Music Information Retrieval (ISMIR)*, pages 550–553, October 2004.
- [17] F. Pachet and J. J. Aucouturier. Improving timbre similarity: How high is the sky. *Journal of negative results in speech and audio sciences*, 1(1):1–13, 2004.
- [18] A. D. Patel and J. R. Iversen. Acoustic and perceptual comparison of speech and drum sounds in the north indian tabla tradition: An empirical study of sound symbolism. In *Proc. of the 15th International Congress of Phonetic Sciences (ICPhS)*, pages 925–928, Barcelona, Spain, 2003.
- [19] R. Smith. An Overview of the Tesseract OCR Engine. In *Proc. of the Ninth International Conference on Document Analysis and Recognition (ICDAR)*, volume 2, pages 629–633, Washington, DC, USA, 2007.
- [20] A. Srinivasamurthy, R. Caro, H. Sundar, and X. Serra. Transcription and recognition of syllable based percussion patterns: The case of Beijing Opera. In *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 431–436, Taipei, Taiwan, October 2014.
- [21] E. Tsunoo, G. Tzanetakis, N. Ono, and S. Sagayama. Beyond timbral statistics: Improving music classification using percussive patterns and bass lines. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(4):1003–1014, 2011.
- [22] R. Typke, F. Wiering, and R. C. Veltkamp. A survey of music information retrieval systems. In *Proc. of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 153–160, London, UK, September 2005.
- [23] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain, D. Kershaw, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland. *The HTK Book, version 3.4*. Cambridge University Engineering Department, Cambridge, UK, 2006.

AUTOREGRESSIVE HIDDEN SEMI-MARKOV MODEL OF SYMBOLIC MUSIC PERFORMANCE FOR SCORE FOLLOWING

Eita Nakamura¹ Philippe Cuvillier² Arshia Cont²
Nobutaka Ono¹ Shigeki Sagayama³

¹ National Institute of Informatics, Tokyo 101-8430, Japan

² Inria MuTant Project-Team, Ircam/UPMC/CNRS UMR STMS, 75004 Paris, France

³ Meiji University, Tokyo 164-8525, Japan

eita.nakamura@gmail.com, philippe.cuvillier@ircam.fr, Arshia.Cont@ircam.fr
onono@nii.ac.jp, sagayama@meiji.ac.jp

ABSTRACT

A stochastic model of symbolic (MIDI) performance of polyphonic scores is presented and applied to score following. Stochastic modelling has been one of the most successful strategies in this field. We describe the performance as a hierarchical process of performer's progression in the score and the production of performed notes, and represent the process as an extension of the hidden semi-Markov model. The model is compared with a previously studied model based on hidden Markov model (HMM), and reasons are given that the present model is advantageous for score following especially for scores with trills, tremolos, and arpeggios. This is also confirmed empirically by comparing the accuracy of score following and analysing the errors. We also provide a hybrid of this model and the HMM-based model which is computationally more efficient and retains the advantages of the former model. The present model yields one of the state-of-the-art score following algorithms for symbolic performance and can possibly be applicable for other music recognition problems.

1. INTRODUCTION

For the last thirty years the real-time matching of music performance to the corresponding score (called score following) has been a popular field of study motivated by applications such as automatic music accompaniment and score-page turning system [1, 2, 3, 4, 5, 6, 7, 8]. We study here score following of polyphonic symbolic (MIDI) performance. A central problem in score following is to properly capture the variety of music performance in a computationally efficient manner. A commonly studied way to capture this variety and develop an effective score-following algorithm is to use stochastic models of music performance (Sec. 2.1, see also [3]).

Hidden Markov models (HMMs) have been applied to score following of symbolic performance and provided currently best results [4, 7, 9]. In these models, a musical event in the score, i.e. note, chord, trill, etc., is represented as a state, and the performed notes are described as outputs of an underlying state transition process. Memoryless statistical dependence is assumed for both output and transition probabilities for the sake of computational efficiency. Due to these simplifications the models cannot well describe significant features of performance data such as the number of performed notes per event and the total duration of a trill.

Phenomenologically, music performance can be regarded as a hierarchical process of producing musical notes: The higher level describes performer's progression in the score in units of musical events, and the lower level describes the production of individual notes [9, 10]. We describe this process in terms of a hidden semi-Markov model (HSMM) [11] with an autoregressive extension [12] (Sec. 2) and incorporate the above features into the model. With some simplifications, the model is reduced to a previously studied HMM [9]. We compare these models in the informational and algorithmic aspects and argue that the present model is advantageous for score following especially for scores with trills, tremolos, and arpeggios (Sec. 3). Empirical confirmation of this fact is given by comparing the accuracy of score following and analysing the errors (Sec. 4). Finally remaining problems and future prospects are discussed (Sec. 5).

2. AUTOREGRESSIVE HIDDEN SEMI-MARKOV MODEL OF SYMBOLIC PERFORMANCE

2.1 Stochastic description of music performance

Music performances based on a score have a wide variety because of indeterminacies inherent in musical score descriptions and uncertainties in movements of performers and musical instruments. These indeterminacies and uncertainties are included in tempos, noise in onset times, dynamics, articulations, ornaments, and also in the way of making performance errors, repeats, and skips [7]. In order to perform accurate and robust score following, we need



© Eita Nakamura, Philippe Cuvillier, Arshia Cont, Nobutaka Ono, Shigeki Sagayama. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Eita Nakamura, Philippe Cuvillier, Arshia Cont, Nobutaka Ono, Shigeki Sagayama. "Autoregressive Hidden Semi-Markov Model of Symbolic Music Performance for Score Following", 16th International Society for Music Information Retrieval Conference, 2015.

to incorporate (maybe implicit) rules into the algorithm to capture this variety.

A way to do this is to construct a stochastic model of music performance and describe those indeterminacies and uncertainties in terms of probability. A score-following algorithm can be developed as an inference problem of the model. We shall take this approach in the following, which has been proved to be successful in score following.

2.2 Model of performer's progression in the score

Let us present the model. We model music performance as a combination of subprocesses in two levels. The higher-level (top-level) process describes the performer's progression in the score in units of musical events that are well-ordered in performances without errors. We take a chord (possibly arpeggiated), a trill/tremolo, a short appoggiatura, or an after note¹ as a unit and represent it with a state (top state). Let i label a top state. Then the performer's progression can be described as successive transitions between these states denoted by $i_{1:N} = (i_1, \dots, i_N)$ (N is the number of performed MIDI notes). We will use the symbol $n (= 1, \dots, N)$ to index the performed notes that are ordered according to the onset time, and i_n represents the corresponding musical event.

The probability $P(i_{1:N})$ describes statistical tendencies of performances. Simplifications are necessary to construct a performance model yielding a computationally tractable algorithm. A typical assumption is that the probability is decomposed into transition probabilities: $P(i_{1:N}) = \prod_{n=1}^N P(i_n | i_{n-1})$ ($P(i_1 | i_0) \equiv P(i_1)$ denotes the initial distribution). The probability $P(j|i)$ represents the relative frequency of straight progressions to the next event ($j = i + 1$), insertions of events ($j = i$), deletions of an event ($j = i + 2$), and repeats or skips (if $|j - i - 1| > 1$). These probability values can be estimated from performance data. With the assumption that $P(i|j)$ is only dependent on $i - j$, the probability values have been estimated with piano performance data in a previous study ([7], Table 3).

2.3 Model of production of performed notes

The lower-level process describes the production of performed notes during each musical event. Because dynamics and articulations are generically highly indeterminate, we focus on pitch and onset time which are denoted by p_n and t_n . For example, multiple notes are performed at a chord or a trill (Fig. 1). Note that whereas chords are written in musical scores as simultaneous notes, performed MIDI notes are serialised and never exactly simultaneous. Thus p_n is always a single pitch.

Let us first consider the number of performed notes per event. For "chords" (meaning a set of all simultaneous notes in the score), short appoggiaturas, and after notes, the expected number of notes is determinate, but it can

¹ Here 'after notes' are defined as grace notes that are played in precedence over the associated beat. A typical example is grace notes after a trill.

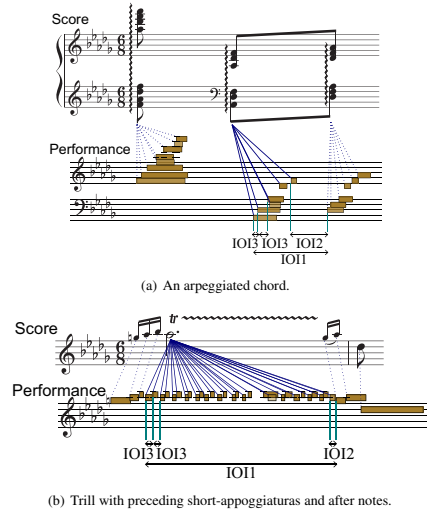


Figure 1. Examples of musical events and performed notes. The three types of time intervals IOI1, IOI2, and IOI3 are explained in the text.

be modified as a result of added or deleted notes by mistake. For trills and (unmeasured) tremolos, the number of notes are indeterminate since the speed of ornaments varies among realisations. We describe this situation with a probability distribution $d_i(s)$ where s denotes the number of performed notes ($\sum_{s=1}^{\infty} d_i(s) = 1$). For example, the function $d_i(s)$ peaks at the indicated number of notes when event i is a chord. When event i is a one-note trill, the peak can be written as $s_i^{\text{peak}} \simeq \nu_i v / \delta t_{\text{trill}}$, where δt_{trill} , ν_i , and v denote the average inter-onset time interval (IOI) of successive notes of a trill, the note value of event i , and the (inverse) tempo in units of "second per unit note value". Because currently we do not have a strong empirical basis for determining the shape of $d_i(s)$, we simply assume it is a normal distribution $d_i(s) = N(s; s_i^{\text{peak}}, \sigma_i)$ with s_i^{peak} given in Sec. 2.3, and leave σ_i as an adjustable parameter.

Next the pitch of each performed note of event i can be described with a probability $P_i^{\text{pitch}}(p)$, which is assumed to be independent for each note for the sake of computational efficiency. The probability values for incorrect pitches represent the possibility and frequencies of pitch errors. An approximate distribution of $P_i^{\text{pitch}}(p)$ has been estimated previously (Eq. (30) of [7]) with piano performance data, where the probability of pitch errors is assumed to be uniform for all score notes.

Finally we consider the description of onset times. A natural assumption of time translational invariance requires the model to be only dependent of time intervals. There are (at least) three different kinds of time intervals

relevant in locally describing onset times of music performance: (IOI1) The time interval between the first notes of succeeding events, which is typically the duration of an event, (IOI2) the time interval between the first note of an event and the last note of its previous event, and (IOI3) the time interval between succeeding performed notes within an event (Fig. 1). Assuming that the probability of these time intervals depends only on the current and previous states for simplicity and computational efficiency, it has the form $P_\kappa(\delta t|i_{n-1}, i_n, v)$ ($\kappa = \text{IOI1, IOI2, IOI3}$) where δt and v denote the relevant time interval and the tempo. Based on the experience that time interval IOI3 is mostly dependent on the relevant event and almost independent of tempo and other contexts, we further simplify the functional form as $P_{\text{IOI3}}(\delta t|i_n)$. Note that the time intervals IOI1 and IOI2 are not independent quantities if we retain all historical information on time, but they have different importance when we take the Markovian description explained below.

2.4 Autoregressive hidden semi-Markov model

The integration of the models in Secs. 2.2 and 2.3 can be described in terms of an extension of the HSMM. In one of equivalent formulations [13] (also Sec. 3.3 of Ref. [11]), a semi-Markov model can be represented as a Markov model on an extended state space. The extended state space is indexed by a pair (i, s) of the top state i (corresponding to a musical event) and a counter of performed notes $s = 1, 2, \dots, 2$ with a transition probability

$$P(i_n, s_n|i_{n-1}, s_{n-1}) = \delta_{s_n,1}P(i_n|i_{n-1})P_{i_{n-1}}^{\text{exit}}(s_{n-1}) + \delta_{s_n, s_{n-1}+1}\delta_{i_n, i_{n-1}}\left(1 - P_{i_{n-1}}^{\text{exit}}(s_{n-1})\right) \quad (1)$$

where

$$P_i^{\text{exit}}(s) = d_i(s)/\sum_{s'=s}^\infty d_i(s'). \quad (2)$$

Here δ in Eq. (1) denotes Kronecker's delta. The exiting probability in Eq. (2) represents the probability that the performer moves to another event given that she has already played s notes at event i . The first term in the right-hand side of Eq. (1) describes the probability that the performer moves to event i_n after having played s_{n-1} notes of event i_{n-1} . The second term describes the probability that the performer stays at event i_n and sounds another note after having played s_{n-1} notes. In this way, this model describes the integrated process of performer's progression in the score and the production of performed notes.

The pitches and onset times of the performed notes can be described with output probabilities associated with this semi-Markov process. We assume the statistical independence of pitch and onset time for simplicity. The output probability of pitch is given by $P(p_n|i_n, s_n) = P_{i_n}^{\text{pitch}}(p_n)$.

The output probability of the onset time of the n -th note

² Remark: In the present model, s counts the number of notes played during a musical event. This is not the durational time (in seconds) spent on that event, which is described with time interval IOI1.

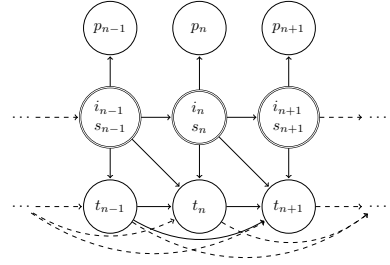


Figure 2. Graphical representation of the autoregressive hidden semi-Markov model of symbolic music performance. The stochastic variables are explained in the text.

is given as

$$P(t_n|i_n, s_n, i_{n-1}, s_{n-1}, v, t_{1:n-1}) = \begin{cases} w_1 P_{\text{IOI1}} + w_2 P_{\text{IOI2}}, & s_n = 1; \\ P_{\text{IOI3}}, & s_n \neq 1 \end{cases} \quad (3)$$

where

$$P_{\text{IOI1}} = P_{\text{IOI1}}(t_n - t_{n-s[n-1]}|i_n, i_{n-1}, v), \quad (4)$$

$$P_{\text{IOI2}} = P_{\text{IOI2}}(t_n - t_{n-1}|i_n, i_{n-1}, v), \quad (5)$$

$$P_{\text{IOI3}} = P_{\text{IOI3}}(t_n - t_{n-1}|i_n)\delta_{i_n, i_{n-1}}. \quad (6)$$

(Here we have written $s[n-1] = s_{n-1}$ to display the equation with clarity.) The three cases correspond to the three kinds of time intervals explained in Sec. 2.3. Because both probabilities for IOI1 and IOI2 have relevance in score following, we have used a mixture probability of them ($w_1 + w_2 = 1$). Such output probabilities with conditional dependence on the previous outputs have been considered in some studies on speech processing, and we call the model autoregressive semi-Markov model based on the convention of previous studies [12]. A graphical representation of the model is given in Fig. 2.

The distributions P_{IOI1} , P_{IOI2} , and P_{IOI3} can be estimated by analysing performance data. The functions P_{IOI2} and P_{IOI3} have previously been estimated with piano performance data [9]. It has been shown there that, in the most important case that $i_n = i_{n-1}+1$ (straight transition to the next event), $P_{\text{IOI2}}(\delta t|i+1, i, v)$ is well approximated by a Cauchy distribution of the form

$$\text{Cauchy}(\delta t; v(\tau_i^{\text{end}} - \tau_i) - \text{dev}_i, 0.4 s). \quad (7)$$

Here $\text{Cauchy}(x; \mu, \Gamma)$ denotes the Cauchy distribution with median μ and width Γ , and τ_i is the onset score time of event i , τ_i^{end} is the score time after which no new onsets of event i can occur, and dev_i describes the ‘stolen time’ of event i whose expectation value is given as the number of short appoggiaturas and arpeggiated notes times the average IOI of the corresponding notes. Using this result, we

can estimate P_{IOI} in the case that $i_n = i_{n-1} + 1$ as

$$P_{\text{IOI}}(\delta t | i+1, i, v) = \text{Cauchy}(\delta t; v\nu_i, 0.4 \text{ s}) \quad (8)$$

where $\nu_i = \tau_{i+1} - \tau_i$ is the note value of event i . The distribution P_{IOI3} was estimated with measurements on IOIs of chordal notes and ornaments (see Secs. 3.3 and 4.2 of [9]).

Finally, tempo v_n is estimated online with a separate model, for which we use a method based on switching Kalman filter (see Sec. 3.4 of [9]). In summary the complete-data probability $P(i_{1:n}, s_{1:n}, t_{1:n}, p_{1:n})$ is given as the following recursive product:

$$\prod_{m=1}^n \left[P(t_m | i_m, s_m, i_{m-1}, s_{m-1}, v_{m-1}, t_{1:m-1}) \cdot P(i_m, s_m | i_{m-1}, s_{m-1}) P_{i_m}^{\text{pitch}}(p_m) \right]. \quad (9)$$

3. COMPARISON WITH OTHER MODELS

3.1 Relation to the HMM-based model

So far the state-of-the-art method for symbolic score following is developed with a performance model based on a standard HMM [9]. The current model can be seen as an extension of this performance model in two ways. First the transition probability of the HMM is realised as a special case of the transition probability in Eq. (1) with exiting probabilities $P_i^{\text{exit}}(s)$ constant in s . Specifically, it is given as the inverse of the expected number of performed notes in event i . As is well known, this constraint leads to a geometrically distributed $d_i(s)$ with a peak at $s = 1$, which is a bad approximation for a large chord or a long trill/tremolo.

The second difference is the structure of output probabilities for onset times. In the standard HMM, the Markovian condition is assumed on the output probability of onset times. Thus the model describes only time intervals IOI2 and IOI3, and the probability distribution for IOI1 in Eq. (3) is ignored. In other words, the IOI output probability of the HMM assumes $w_1 = 0$ and $w_2 = 1$ in that equation. This means that the total duration of a trill/tremolo or an arpeggio is poorly captured with the HMM.

These differences have important effects when the models are applied to score following. For score following, the pitch information is generically most important. When there are musical events with similar pitch contents in succession, however, the information on onset times and the number of performed notes play more significant roles in correctly matching notes. For example, to correctly match performed notes of succeeding trills/tremolos, the number of notes and the duration of each trill/tremolo are important viewpoints. Since they are not well captured in the HMM, the autoregressive HSMM would work better in this case. Similar situations arise for successions of arpeggios, where the time intervals IOI2 and IOI3 are largely variable among realisations. On the other hand, the time intervals IOI1 and IOI2 are almost same for successive normal chords and these IOIs carry much information necessary to cluster them. Thus the models are expected to have similar effects for passages without ornaments.

3.2 Comparison with the preprocessing method

To solve the problems with ornaments for score following, a preprocessing method has been proposed long ago [14]. The idea is to preprocess performed notes so that ornamental notes are not sent to the matching module directly. While the method can work for scores with not-heavy polyphonic ornamentation and performances with infrequent errors, the preprocessing can fail when there are errors or unexpected repeats or skips near ornaments. Because a direct comparison showed that the HMM outperformed the preprocessing method for piano performances with errors, repeats, and skips [9], we compare our model only with the HMM in Sec. 4.

3.3 Computational cost

For score following, we find the most probable hidden state sequence given the input performance. In order to realise real-time processing, the computational cost of the estimation algorithm must be sufficiently small. We here compare the present model and the HMM discussed in Sec. 3.1 in terms of the computational cost.

The Viterbi algorithm can be applied for HMMs to estimate states. Let us denote the product of the transition probability and the output probability as $a_{ij}(o) = P(j|i) \cdot P(o|i, j)$ where o represents pitch and onset time. The Viterbi update equation can be expressed as the following recursive equation

$$\begin{aligned} \hat{p}_N(i_N) &\equiv \max_{i_1, \dots, i_{N-1}} \left[\prod_{n=1}^N a_{i_{n-1}i_n}(O_n) \right] \quad (10) \\ &= \max_{i_{N-1}} [\hat{p}_{N-1}(i_{N-1}) a_{i_{N-1}i_N}(O_N)]. \quad (11) \end{aligned}$$

The number of states is N since a state corresponds to a musical event in the score. If we allow arbitrary progressions in the score including repeats and skips, a direct application of the Viterbi algorithm requires $\mathcal{O}(N^2)$ computations of probability for each update. When the probability matrix $a_{ij}(o)$ can be represented as a sum of a band matrix α_{ij} of width D and an outer product of two vectors S_i and r_j , the computational complexity can be reduced to $\mathcal{O}(DN)$ with a recombination method [7]. Intuitively, α_{ij} describes probabilities corresponding to transitions between neighbouring states, which have larger probabilities, and S_i and r_j represent probabilities corresponding to large repeats and skips, which typically have very small probabilities. Substituting $a_{ij}(o) = \alpha_{ij} + S_i r_j$ into Eq. (11), we see α_{ij} induces $\mathcal{O}(DN)$ complexity and $S_i r_j$ induces $\mathcal{O}(N)$ complexity by a recombination. This simplified transition probability matrix is used in previous studies to enable real-time processing for long scores.

It is clear from the formulation of the autoregressive HSMM in Sec. 2.4 that the standard Viterbi algorithm can also be applied to the model. In practice, we put an upper bound on the number of performed notes s_i^{\max} for each event i , and the number of states of the HSMM is $\sum_i s_i^{\max} \equiv SN$ where S is the average of s_i^{\max} . Because of the special form of transition probabilities in Eq. (1), the

Table 1. Error rates (%) of score following with the autoregressive HSMM (“HSMM”), the hybrid model (“Hybrid”), and the HMM [9]. The first four pieces indicate Couperin’s *Allemande à deux clavecins*, the solo piano part of Beethoven’s first piano concerto, Beethoven’s second piano concerto, and Chopin’s second piano concerto [9], and the last two pieces are explained in the text.

Piece	# Notes	HSMM	Hybrid	HMM
Couperin	1763	5.50	6.02	6.66
Beethoven 1	17587	3.16	3.13	3.16
Beethoven 2	5861	2.01	2.20	2.35
Chopin	16241	9.22	9.22	11.1
Debussy	3294	3.64	3.58	4.66
Tchaikovsky	2245	0.40	0.40	4.55

computational complexity for one Viterbi update is generically $\mathcal{O}(SN^2)$. When we apply the recombination method in Ref. [7], the complexity can be reduced to $\mathcal{O}(DSN)$ for the outer-product type transition probability. Note that the width D in the top-level transition probability matrix induces SD transitions between HSMM states. Consequently the computational cost of the model is about S times larger than its reduced HMM. For example, if we set s_i^{\max} as twice the number of expected notes per event, $S \simeq 3-10$ for a score with a modest degree of polyphony, and it increases if there are many large chords or long trills/tremolos.

3.4 Hidden hybrid Markov/semi-Markov model

As discussed in Sec. 3.1, there are reasons that the present model yields better results for score following than the HMM, but it is at the cost of increased computational cost, which is unwanted for long scores. On the other hand, most of the musical events in scores are normal chords (or single notes) for which the HMM already yields good results. Therefore if we combine the HMM state representation for normal chords and the autoregressive HSMM state representation for other ornamented events, it would be possible to obtain an improved score-following algorithm with minimal increase in computational cost. Such a combination of HMM and HSMM can be achieved in the framework of hidden hybrid Markov/semi-Markov model [5, 15]. In the hybrid model, normal chords are represented with HMM states and other events (i.e. trill, tremolo, arpeggio, short appoggiatura, and after notes) are represented with HSMM states. For this model the computational complexity of the Viterbi algorithm takes the same form as the autoregressive HSMM, by substituting $s_i^{\max} = 1$ for HMM states in $S = \sum_i s_i^{\max}/N$.

4. COMPARING THE ACCURACY OF SCORE FOLLOWING

To evaluate and compare the discussed models with respect to the accuracy of score following, we implemented

Table 2. Number of mismatched notes of various types. Each type is explained in the text. The same abbreviations for the models as in Table 1 are used.

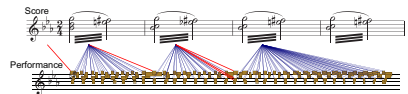
Type	# Notes	HSMM	Hybrid	HMM
Trill	8159	282	281	508
Tremolo	2603	115	115	151
Arpeggio	1081	36	33	127
Other ornaments	2401	340	339	362
Other	32030	1580	1599	1673

three score-following algorithms based on the autoregressive HSMM (Sec. 2.4), the hybrid model (Sec. 3.4), and the HMM [9], and run these algorithms for music performance data containing various ornaments. In addition to the piano performance data used in Ref. [9] which contains performance errors, repeats and skips, we used collected piano performances of passages in Debussy’s *En Blanc et Noir* with successions of tremolos (the first piano part in the second movement) and the solo piano part of Tchaikovsky’s first piano concerto with his typical successions of wide arpeggios (the last section of the second movement).

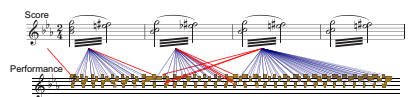
The additional parameters σ_i for the autoregressive HSMM and the hybrid model were set as follows: $\sigma_i = 0.4s_i^{\text{peak}}$ for trills and tremolos and $\sigma_i = 1$ otherwise. The mixture weights for the output probability for time intervals IOI1 and IOI2 were set as $w_1 = w_2 = 1/2$. These parameters were used as a benchmark and there is a room for further optimisation.

For the evaluation measure, we calculated the error rate, which is defined as the proportion of mis-matched notes to the total number of performed notes. There were performed notes that are difficult to associate with any score notes even for humans, which naturally appear in real data. While they were included in the input data, they were not used in the calculation of error rates. Results are shown in Table 1, where we see that the autoregressive HSMM and the hybrid model had similar accuracies, and the HMM had the worst accuracy overall. (Slight differences in the values for the HMM compared to those in Ref. [9] are mainly due to slight corrections of the implementation.) For detailed error analysis, we list the frequencies of classified matching errors in Table 2. Here the numbers indicate the total number of matching errors in the whole data for each type. Ornaments are classified into the first four types, and other notes are gathered in the last type. Significant reduction of matching errors is observed in the first three types (trill, tremolo, and arpeggio), and other types of matching errors are also reduced but rather slightly in the reduction rate.

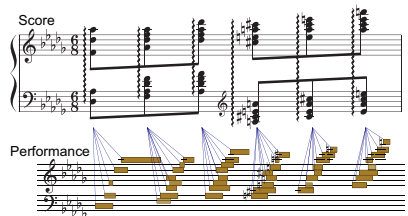
Two example results of score following are shown in Fig. 3, which represent typical situations where the autoregressive HSMM worked better than the HMM. In the first example, the passage includes a succession of tremolos with similar pitch contents. We see some of the mismatched notes with the HMM are correctly matched with the autoregressive HSMM. Similarly the mismatched notes



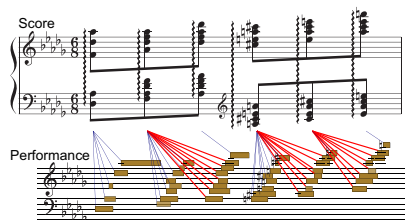
(a) A passage from Debussy's *En Blanc et Noir* with the autoregressive HSMM.



(b) Same as (a) with the HMM.



(c) A passage from Tchaikovsky's first piano concerto with the autoregressive HSMM.



(d) Same as (c) with the HMM.

Figure 3. Example results of score following with the autoregressive HSMM and the HMM [9]. Mismatched notes are indicated with bold red lines.

Table 3. Averaged computation time (ms) required for one Viterbi update. The same abbreviations for the models and the musical pieces as in Table 1 are used.

Piece	HSMM	Hybrid	HMM
Couperin	1.6	1.1	0.3
Beethoven 1	5.9	2.9	1.1
Beethoven 2	7.0	3.0	1.6
Chopin	7.1	3.5	1.2
Debussy	0.9	0.8	0.1
Tchaikovsky	1.2	1.0	0.1

with the HMM are all correctly matched with the autoregressive HSMM for a succession of wide arpeggios in the second example. These results are consistent with the discussion in Sec. 3.1.

We also measured the required computation time (Table 3). The computation time for each Viterbi update is constant over time, and the algorithms were run on a laptop with moderate computation power. The results confirm our expectation that the use of hybrid model for score following has practical advantages over the autoregressive HSMM in the computation time and the HMM in the accuracy.

5. CONCLUSION

We explained reasons that the present model of symbolic music performance based on autoregressive HSMM is more advantageous for score following than previously studied HMMs, and we have confirmed this empirically by comparing the accuracy of score following and analysing the matching errors. Because a semi-Markov model can be seen as a Markov model with an extended state space as we have explained, we can apply to the present model the methods for HMMs to improve score following [7, 16]. In particular, this is important to reduce matching errors occurring after repeats and skips and those due to reordered notes in the performance, which were the main factors of remaining errors.

It would be interesting to apply the present model for music/rhythm transcription and related problems. Because the model describes both the total duration and the internal temporal structure of ornaments, it would be possible to detect ornaments from performances without a score and integrate the results into music transcription.

6. ACKNOWLEDGEMENTS

This work is partially supported by NII MOU Grant in fiscal year 2014 and Grant-in-Aid for Scientific Research from Japan Society for the Promotion of Science, No. 26240025 (S.S. and N.O.) and No. 25880029 (E.N.).

7. REFERENCES

- [1] R. Dannenberg, "An on-line algorithm for real-time accompaniment," *Proc. ICMC*, pp. 193–198, 1984.
- [2] B. Vercoe, "The synthetic performer in the context of live performance," *Proc. ICMC*, pp. 199–200, 1984.
- [3] N. Orio, S. Lemouton and D. Schwarz, "Score following: State of the art and new developments," *Proc. NIME*, pp. 36–41, 2003.
- [4] B. Pardo and W. Birmingham, "Modeling form for on-line following of musical performances," *Proc. of the 20th National Conf. on Artificial Intelligence*, 2005.
- [5] A. Cont, "A coupled duration-focused architecture for realtime music to score alignment," *IEEE Trans. PAMI*, **32(6)**, pp. 974–987, 2010.
- [6] A. Arzt, G. Widmer and S. Dixon, "Adaptive distance normalization for real-time music tracking," *Proc. EU-SIPCO*, pp. 2689–2693, 2012.
- [7] E. Nakamura, T. Nakamura, Y. Saito, N. Ono and S. Sagayama, "Outer-product hidden Markov model and polyphonic MIDI score following," *JNMR*, **43(2)**, pp. 183–201, 2014.
- [8] P. Cuvillier and A. Cont, "Coherent time modeling of semi-Markov models with application to real-time audio-to-score alignment," *Proc. IEEE MLSP*, 6 pages, 2014.
- [9] E. Nakamura, N. Ono, S. Sagayama and K. Watanabe, "A stochastic temporal model of polyphonic MIDI performance with ornaments," to appear in *JNMR*, 2015.
- [10] N. Orio and F. Déchelle, "Score following using spectral analysis and hidden Markov models," *Proc. ICMC*, pp. 1708–1710, 2001.
- [11] S.-Z. Yu, "Hidden semi-Markov models," *Artificial Intelligence*, **174**, pp. 215–243, 2010.
- [12] J. Bilmes, "Graphical models and automatic speech recognition," in *Mathematical foundations of speech and language processing* (Springer New York), pp. 191–245, 2004.
- [13] M. Russel and A. Cook, "Experimental evaluation of duration modelling techniques for automatic speech recognition," *Proc. ICASSP*, pp. 2376–2379, 1987.
- [14] R. Dannenberg and H. Mukaino, "New techniques for enhanced quality of computer accompaniment," *Proc. ICMC*, pp. 243–249, 1988.
- [15] Y. Guédon, "Hidden Hybrid Markov/Semi-Markov Chains," *Computational Statistics and Data Analysis*, **49**, pp. 663–688, 2005.
- [16] E. Nakamura, Y. Saito, N. Ono and S. Sagayama, "Merged-output hidden Markov model for score following of MIDI performance with ornaments, desynchronized voices, repeats and skips," *Proc. Joint ICMC|SMC 2014*, pp.1185-1192, 2014.

AUTOMATIC MASHUP CREATION BY CONSIDERING BOTH VERTICAL AND HORIZONTAL MASHABILITIES

Chuan-Lung Lee¹ Yin-Tzu Lin¹ Zun-Ren Yao¹
Feng-Yi Lee² Ja-Ling Wu¹

Communications and Multimedia Laboratory, National Taiwan University, Taiwan

¹{kane0986, known, yyy110011, wjl}@cmlab.csie.ntu.edu.tw, ²milkycc1111@gmail.com

ABSTRACT

In this paper, we proposed a system to effectively create music mashups – a kind of re-created music that is made by mixing parts of multiple existing music pieces. Unlike previous studies which merely generate mashups by overlaying music segments on one single base track, the proposed system creates mashups with multiple background (e.g. instrumental) and lead (e.g. vocal) track segments. So, besides the suitability between the vertically overlaid tracks (i.e. vertical mashability) used in previous studies, we proposed to further consider the suitability between the horizontally connected consecutive music segments (i.e. horizontal mashability) when searching for proper music segments to be combined. On the vertical side, two new factors: “harmonic change balance” and “volume weight” have been considered. On the horizontal side, the methods used in the studies of medley creation are incorporated. Combining vertical and horizontal mashabilities together, we defined four levels of mashability that may be encountered and found the proper solution to each of them. Subjective evaluations showed that the proposed four levels of mashability can appropriately reflect the degrees of listening enjoyment. Besides, by taking the newly proposed vertical mashability measurement into account, the improvement in user satisfaction is statistically significant.

1. INTRODUCTION

A Mashup is a kind of popular music what is made by overlaying, connecting, digitally modifying parts of two or more existing audio recordings [29]. The most common way to create a mashup is to overlay the vocal track of one song on the instrumental track of another [29]. With the aid of high-speed Internet, users are more easily to trade music materials and find related information through social websites [1, 3]. The development and availability of digital audio editing techniques and software also reduced the entry barrier for creating mashups. For example, the

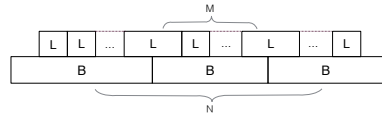



Figure 1. Common structure of mashup songs. Each block labeled with “L” or “B” represents a segment in lead track (e.g. vocal track) or the background track (e.g. instrumental track) from the same songs, respectively. M and N denote the number of lead track segments per background segment and the total number of background track segments in the resultant mashup, respectively.

loop-based music sequencers such as Sony ACID Pro and Ableton Live make it easier for users to match the beats and shift the keys of the audio samples. As a result, mashups are now more often created by music lovers without formal musical training [29]. However, with the aforementioned tools, users still need to rely on their own experiences and musical training to find out proper music clips to be combined together. As the amount of currently available digital music explosively goes up, finding suitable clips becomes time-consuming and labor-intensive. How to automatically find out and create pleasant mashups becomes a challenging and interesting issue.

Some previous studies have proposed automatic schemes to create mashups. But those approaches [8, 9] merely focused on the vertical suitability of the chosen music segments, that is, “they considered only about how suitable are the music segments to be overlaid”, which was defined as the term “mashability” in [8]. By observation, many human made mashups¹ are not created just by overlaying different music segments on one single base track, as proposed in [9]. A Mashup can also be composed of segments of multiple background tracks (e.g. instrumental tracks) segments from different songs. Each background track segment is overlaid with several lead track segments (e.g. vocal tracks). As shown in Figure 1, when the background track segments changed, the lead tracks on top of them may still remain in the same song. So, while finding proper segments for generating mashups, we need to consider not only the vertical mashability between lead and background track segments but also the horizontal relation-

 © Chuan-Lung Lee, Yin-Tzu Lin, Zun-Ren Yao, Feng-Yi Lee and Ja-Ling Wu. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Chuan-Lung Lee, Yin-Tzu Lin, Zun-Ren Yao, Feng-Yi Lee and Ja-Ling Wu. “Automatic Mashup Creation by Considering Both Vertical and Horizontal Mashabilities”, 16th International Society for Music Information Retrieval Conference, 2015.

¹ <https://www.youtube.com/watch?v=IF5MF4wm1T8>

ships between consecutive lead/background segments – we defined this relation as the “horizontal mashability”.

In this work, a framework is proposed to automatically create mashups by considering both the vertical and the horizontal mashabilities. Besides, two additional factors: “harmonic change balance” and “volume weighting” to the vertical mashability are also considered and investigated. Subjective evaluation shows, by taking these factors into account, the users’ listening pleasure of the created mashups will be enhanced as compared with that of the original counterparts created in [9]. Moreover, by integrating with the horizontal mashability, various degrees of listening enjoyment of mashups can be achieved. As a result, given a set of multitrack songs with structural segment labels, the first background track users want to extract from and some desired structure factors (such as the number of background track segments N , and the number of lead track segments per background segment M), the system will then automatically generate a pleasant mashup with the structure as illustrated in Figure 1.

We assume that the multitrack songs should at least contain two kinds of tracks: background and lead. This assumption is reasonable because multitrack songs can be easily retrieved from mashup-related social websites [1, 3]. The unit of input segments depends on the granularity of user specified song changes in a mashup. The unit could be as large as a structural section (e.g. verse, chorus), or be as small as a musical phrase (e.g. half or quarter of a verse), but we assume that all segment boundaries are aligned with bars. If users are not willing to provide segment boundaries, we can still detect the boundaries by using current structural segmentation techniques [14]. These input segments are regarded as the basic units to create mashups. To distinguish “input segment” from the generally used term “segment”, in the rest of the paper, we will term it as “unit”. Therefore, in this paper, a lead unit stands for a segment in the lead track of an input song, and so on.

2. RELATED WORK

As compared with other music genres, mashup music is still young, so there is still a few academic studies focused on automatic mashup creation. Griffin et al. [13] proposed an efficient way to adjust the tempi of user-specified tracks and combine them after synchronizing their beats. The commercial software – Mixed in Key Mashup [2] uses the global harmonic compatibility among tracks in the users’ music collection as the cue for track screening and provides tools to help users match the beats of the chosen tracks. In other words, users still need to find out proper segments in the chosen tracks by themselves. AutoMashupper [8, 9] is the first study that provided a thorough investigation on measurement for finding proper music segments to be overlaid together and an automatic mashup generation scheme. In AutoMashupper [9], an input song is regarded as a base track, and is segmented into short segments. For each segment, segments from other songs that are with the highest mashability—on the basis of chromagram similarity, rhythmic similarity, and spectral

balance – will be overlaid with the corresponding segment in the base track to create the final mashup. The subsequent studies [7, 27] also followed this structure. In [7], a live input audio is regarded as the base track, and the accompanied music segments are overlaid upon the input audio. Tsuzuki et al. [27] focused on helping users overlay voices from different singers who had sung the same song along the common accompanied track. The proposed system, in contrast, is capable of creating mashups from multiple background and lead segments.

Besides mashup creation, there are other studies focused on mixing parts from existing music recordings, by means of concatenating instead of overlaying the music segments, such as the automatic DJ [6, 15] [16, p. 97-101], the medley creation systems [18, 20] and concatenative synthesis [4, 24] [16, p. 101-102, 109-111]. The former two types of studies focused on concatenating longer audio segments such as phrases or sections, and studies of concatenative synthesis focused on audio snippets that are as short as musical notes/onsets. To select proper units to be concatenated, the existing systems may pick up proper candidates by comparing the similarity/distance between the candidates and the given unit according to various audio features (e.g. tempo, rhythm, pitch, harmonic, and timbre) [15, 16, 18], pre-cluster the all the units and then choosing among them according to some statistical models [4, 20, 24], or align them with user specified conditions [4, 6, 20]. For short units (e.g. notes), the units may be concatenated directly or accompanied with short cross-fade. For long units (e.g. sections or phrases), the above-mentioned systems may first decide the transition positions between consecutive music segments on the basis of rhythm [16] or chroma [18] similarity. And then, they adjusted the tempi (e.g. by phase vocoder [12]) and aligned the beats in the music segments with various methods and then concatenated the segments by cross-fading. In this study, the pre-described methods used to find proper segments and to smoothly connect them will be well-incorporated in the horizontal stage of the proposed system.

3. PROPOSED FRAMEWORK

The proposed system framework is illustrated in Figure 2. In the preprocessing step, the system will first extract audio features and pre-compute vertical and horizontal mashabilities for each possible pair of units in the given music set. Then, according user specified structure factors (e.g. the first song, the number of background track segments N , the number of lead track units per background segment M , etc.), we will determine (i) which and where the audio segment should locate in the resultant mashup – mashup composition (ii) how these segments are transformed to generate the resultant mashup. – mashup generation. In the “mashup composition” step, we will first pick M consecutive background units from the user specified song. We termed these units as a group of background unit (GBU). If users did not specify the first song, our system will randomly choose a GBU for them. Then, in the verti-

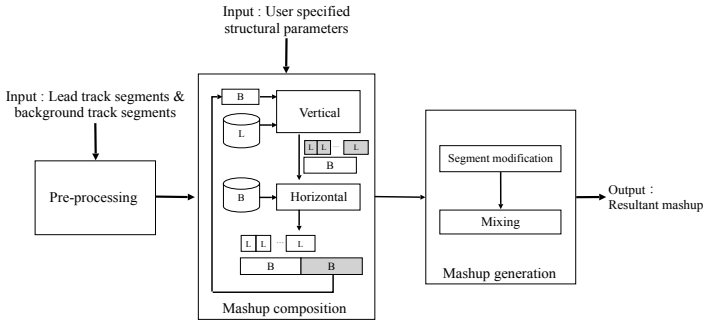


Figure 2. Proposed system Framework.

cal stage, we focus on finding proper lead units (the gray blocks marked with “L” in Figure 2) to be overlaid with the input GBU via vertical multiple mashabilities. After that, in the horizontal stage, we aim at finding a proper subsequent GBU (the gray block marked with “B” in Figure 2) by considering both vertical and horizontal mashabilities. The two processes, vertical and horizontal mashup stages, will be run iteratively until the resultant mashup reaches user desired length – the number of GBU N . Finally, in the mashup generation step, the tempo, loudness and pitch of each unit will be first modified to the desired values, and then the units will be mixed and concatenated to generate the final mashup song.

4. PREPROCESSING

In this step, the system will first extract audio features and pre-compute vertical and horizontal mashabilities for each possible pair of music units. The used features are beat/tempo [11], beat-synchronous chromagram [21], chord [22], MFCC [10], and volume [23]. For the ease of understanding, we will describe the used mashabilities, and how the above mentioned features are combined with each mashability in the following sections.

5. MASHUP COMPOSITION

In mashup composition, our system will determine which and where the basic units should locate. First, we will pick a GBU as our starting point (user specified or randomly picked by the system). The GBU should be with M consecutive background units in a song. Besides, all the background units in a GBU should contain exactly 2^κ beats, for $\kappa \in \mathbb{N}$, $\kappa \geq 2$. The reasons are (i) most popular songs are in 4/4 meter – 4 beats in a bar. (ii) most musical phrases in pop songs are multiples of four bars long [28]. (iii) most verse or chorus sections contain 2 to 4 phrases [28].

5.1 Vertical Stage

In the vertical stage, our system will find proper multiple lead units for each of the background unit in the input

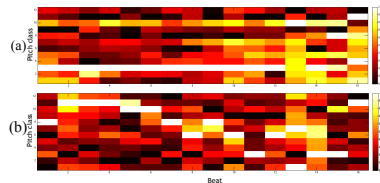


Figure 3. Chromagrams of the segment unit with (a) simple texture and (b) complex texture .

GBU based on vertical mashabilities. Mashabilities used in previous studies [9] include, harmonic matching, rhythmic matching and spectral balance. We do not use rhythmic matching in this stage because most lead tracks have no kick or snare sounds, so rhythmic pattern becomes unreliable in finding lead units. Spectral balance is also eliminated because the sounds in lead tracks often spread in the mid-band (220-1760 Hz), then spectral balance becomes indistinguishable. As a result, we adopt harmonic matching, and propose two new vertical mashabilities: harmonic change balance, and volume weighting.

5.1.1 Harmonic Matching

In harmonic matching part, we use a similar method to that of the AutoMashUpper [9]. The major difference is that we directly calculate the chroma similarity between each lead unit and background unit instead of shifting a window in the whole song. The reason is that the original method can not guarantee to get a complete lead unit. This may cause problems for the subsequent horizontal stage process, especially when it is the last unit in a GBU, because we will need to find consecutive lead units near GBU boundaries (please refer to Section 5.2 for details). The mashability score calculated by harmonic matching is denoted as S_c .

5.1.2 Harmonic Change Balance Weighting

Harmonic change balance is a newly proposed mashability. The idea comes from the observation that chroma similar-

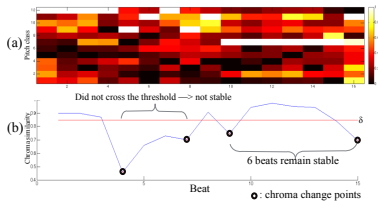


Figure 4. (a) Chromagrams of a segment unit and (b) the corresponding plot of chroma similarity between consecutive beats.

ities are not always proportional to the suitability for overlaying two segments. For instance, a given GBU composed of only one long note or long chord, the highest chroma-similar lead unit to it will highly probably be lead units that are also composed of the same texture. Then, the picked lead units for the GBU will all sound alike – long notes or long chords, which makes the resultant mashup sound boring and meaningless, even it sounds quite harmonic because of high chroma-similarity. As a result, we proposed to match the input unit to the one that is composed of opposite harmonic change rate, e.g., a background unit with simple texture (such as the chromagram illustrated in Figure 3(a)) should match with a lead unit with complex texture (c.g. Figure 3(b)), and vice versa. The harmonic change rate can be calculated according to how many beats remain stable on the chroma in a unit. Figure 4 illustrates the chromagram and the chroma similarities between consecutive beats in a unit. The local minima of the chroma similarity plot below the threshold δ can be defined as the chroma change points. Then, the beats lie between any two change points and contain exactly two crossing points to δ are regarded as stable beats. The percentage of stable beats will be mapped to a sigmoid function to get a smooth score from 0 to 1 (0% stable beat is mapped to 1 while, 100% stable beats are mapped to 0), i.e. the harmonic change rate ξ . Then, the harmonic change balance weights w_t can be calculated as:

$$w_t = 1 - |\xi_p - (1 - \xi_q)|, \quad (1)$$

where ξ_p and ξ_q are the harmonic change rate of units p and q , respectively. If harmonic change rate of a background unit is 0.7, we tend to find a lead unit whose harmonic change rate is closer to 0.3.

5.1.3 Volume Weighting

There are many inaudible (less than -40db) lead units in a lead track because the lead vocal or instruments often rest in sections such as intro, intermezzo, and outro. To eliminate lead units contain too many inaudible parts, we included the volume weighting w_v in our vertical mashability computation. w_v can be calculated according to the portion of the lead units that can be heard. That is,

$$w_v = \begin{cases} 1 & , \text{ if } a \geq \frac{1}{2}\eta \\ \frac{1}{2} + \frac{a}{\eta} & , \text{ if } a < \frac{1}{2}\eta, \end{cases} \quad (2)$$

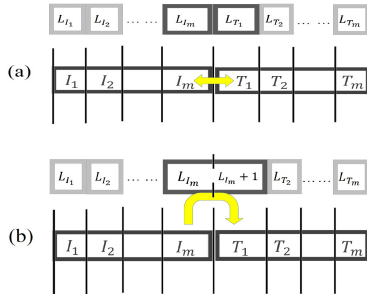


Figure 5. Schematic diagrams showing how to generate mashups by considering (a) horizontal mashability and (b) vertical mashability, respectively.

where a and η are the numbers of audible and total beats in a unit, respectively.

Finally, we combine the aforementioned measurements together to find the final vertical mashability S_v , that is

$$S_v = S_c \cdot w_t \cdot w_v + w_\tau, \quad (3)$$

where w_τ is an additional bonus to the pair of units with close tempo, it is similar to the parameter α adopted in Eqn. (9) of [9].

5.2 Horizontal Stage

In this stage, we aim at finding a proper subsequent GBU **T** for the input GBU **I** by considering both vertical and horizontal mashabilities. A perfect subsequent GBU **T** should satisfy two properties: (i) it can smoothly be concatenated with the previous GBU **I** (cf. Figure 5 (a)), and (ii) one can find a proper leader unit on top of the first background unit in this GBU **T** and the found leader unit can be smoothly concatenated with the previous leader unit (cf. Figure 5 (b)). To achieve property (i), we incorporated an approach similar to the concept described in [20, Sec. 7.2] and [19, Sec. 4.1]. We adopted the same similarity measurements and weights as [20] to compute the similarity between the GBU subsequent to **I** in the original track and all the candidate GBUs, which was defined as the horizontal mashability S_h . Then, sort according to S_h , we can get a rank list, R_h . A threshold α is applied to cut off the rank list: the GBUs with S_h that are lower than α are eliminated. For dealing with the pre-described property (ii), we take the opposite direction. That is, we first check if the next unit in the original track of the lead unit L_{i_m} exists. If it is, we will temporally choose it as the first lead unit for GBU **T**. Then, we can get another rank list R_v of GBUs by sorting the vertical mashabilities S_v between the first background units of the GBUs and the picked lead unit. A similar threshold β is also applied to the rank list to eliminate inappropriate GBUs. After the above steps, we may encounter four cases in the transitions between two GBUs.

- Case 1. Both R_h and R_v exist, and $\{R_h \cap R_v\} \neq \emptyset$. This is the perfect case. Then, we can pick the first GBU in $\{R_h \cap R_v\}$ as our result.
- Case 2. Only R_v exists. We pick the first GBU in R_v . In case 2, the two background units at the transition have no correlation but they are bridged via the lead units taken from the same song on top of them.
- Case 3. The opposite of case 2. Only R_h exists. We choose the first GBU from R_h . In this situation, two background units at the transition have high correlation but the lead units on top of them cannot stay in the same track.
- Case 4. Both R_h and R_v do not exist. We randomly choose a GBU. In case 4, the two background units have no correlation and the lead units on top of them cannot stay in the same track. We can also provide an optional self-repairing mechanism for case 4. That is, instead of random selection, we choose a GBU that its next transition will fit the condition of case 1 via pre-computation of all the possible cases of all the GBUs in the collection.

A more complex situation is that, both R_h and R_v exist, but $\{R_h \cap R_v\} = \emptyset$. Which rank list should we choose from? According to the user evaluation results in Section 7.2, most users prefer case 2 than case 3. So we will choose a GBU from R_v first.

6. MASHUP GENERATION

Mashup generation can be divided into two steps: segment modification and mixing. In segment modification, we first shift the pitches of the lead units to a target key, found in the harmonic matching step (Section 5.1.1). The same as [9], we use Rubberband library [5] to shift the pitches. Then, the volume of the lead units are also re-scaled to match that of the background unit by Replay Gain [23]. After that, to match to beats of the units, we apply phase vocoder [12] to stretch the beats. Finally, we extend all of the units one beat long and apply cross fade technique to all of the transitions to create the resultant mashup.

7. EXPERIMENT: SETTINGS AND RESULTS

We conducted two subjective listening tests. The first test is to evaluate the impact and the user's acceptability of the four approaches, we proposed to deal with various transition conditions and also find the proper-connecting priority of these four cases. The second test is to compare the compatibilities of lead units which are provided by the mashability in AutoMashUpper [9] and by the vertical mashability in our system. The generated mashups can be found in <http://cmlab.csie.ntu.edu.tw/~kane0986/ISMIR2015.html>.

7.1 Dataset

We use the multi-track audio dataset in [14] with structural segment labels. The dataset contains 104 pop songs, each

song contains about 5 tracks on average. In the experiment, for each song, we take the lead vocal track as the lead track, and then we mix all the rest tracks into a single track and regard it as the background track. Examples of background tracks are drum and bass tracks or chordal instruments such as piano, guitar, or string. The vocal chorus track is eliminated because it has different properties to either the lead or the background track. We take 0.8478 as the threshold δ in the harmonic change balance weighting step. The threshold is obtained by finding the intersection of distribution of the chroma similarity values of consecutive beats in 73 simple textured units and 156 complex units. The other two thresholds, α and β we used in the horizontal stage are set to 0.6422 and 0.5740, respectively. These two thresholds are obtained through observations on the first derivatives of the sorted scores of all the unit pairs.

7.2 Subjective Evaluations on Horizontal Mashability

In the first test, given the same background unit \mathbf{B} and lead units on top of \mathbf{B} as inputs, we then got four mashups which have dedicated configurations as those of pre-described case 1 to case 4, respectively. We also added the original track unit \mathbf{B} , which has no transition, as a reference, and is denoted as case 0. The user evaluations are conducted through the aid of a web interface, and the tested mashups are presented in random order. For each participant, he or she needs to listen five groups of mashups, and each group contains five mashups which respect to the five cases we mentioned above. The questionnaires are designed based on a 7-point Likert scale [17]. Users are asked to report their opinions about the degrees of enjoyment of the mashups from the following options: very pleasing (7), pleasing (6), somewhat pleasing (5), neutral (4), not so pleasing (3), not pleasing (2), and very unpleasing (1). 21 males and 6 females aged around 20~60 participated in this test. All of our participants have listening test experience, but most of them are not major in music (less than five participants have educational background in music) since our target consumer is general public.

Figure 6 shows the mean opinion score of each case. The paired Wilcoxon signed rank test [25] is applied to analyze the results, where the corresponding p-values are reported in Figure 6. The overall result shows that the four proposed cases did impact the human feeling of the resultant mashups under a confidence level of 95%². Case 1 is rated second only to case 0. The score of case 2 is lower than case 1, but commonly higher than case 3. This indicates that users commonly prefer case 2 to case 3, i.e., bridging two GBUs with no relation via one lead track is more acceptable than concatenating two GBUs with high correlation but with the lead units do not stay in the same track. Case 3 is rated higher than case 4 commonly, but not as significant as other cases. Case 4 gets the lowest score generally, this verifies that when there is significant change in both of the lead and the background transitions, a great impact to user's acceptability will result.

² By Bonferroni correction, to preserve the total confidence level as 95%, the p value for each paired comparison should be $< \frac{0.05}{C_2^4} = 0.005$

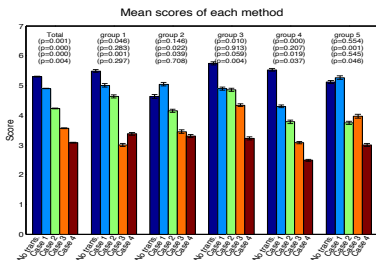


Figure 6. Mean opinion scores of total and each test sample, in which the relevant p-values of paired Wilcoxon signed rank test [25] on “case 0 (no transition) vs. case 1”, “case 1 vs. case 2”, “case 2 vs. case 3”, and “case 3 vs. case 4” are displayed above the corresponding bars of each one of the experiments, respectively.

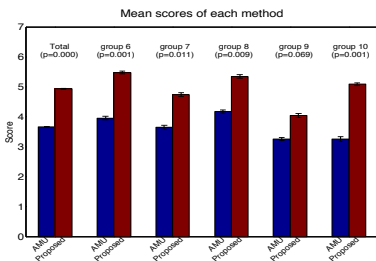


Figure 7. Mean opinion scores of total and each test sample, in which the relevant p-values of paired Wilcoxon signed rank test [25] on “AMU vs. our system” are displayed above the corresponding bars of each experiment.

Counter-intuitively, there are two groups in our listening test showing that case 1 is rated higher than case 0 slightly though they did not statistically significant. Possible reason would be that the two GBUs in case 0 happen to be from verse and chorus sections of different styles, respectively. Then our system may have chance to find another background unit which can be concatenated after the verse segment more smoothly than it’s own chorus counterpart.

7.3 Subjective Evaluations on Vertical Mashability

In the second test, we aim at comparing the vertical mashability provided by our system and the AutoMashUpper [9] (denoted as AMU). We create the mashups by our method and AMU’s method from the same input GBU I of 6 background units. Besides, we force the chosen lead units to be picked from different songs. As we mentioned in Section 5.1, rhythmic matching and spectral balance are not reliable for the current dataset, so we only used the harmonic matching part in AMU³, i.e. the version in [8].

³ We implemented AMU’s methods by ourselves.

Then, it is easily to pick lead units that are nearly inaudible since only harmonic matching is considered in the adopted AMU version. To make a fair test, we also apply our volume weighting (Section 5.1.3) to AMU. As a result, the target component we compared here is the harmonic change balance weighting. A similar evaluation procedure to the previous experiment was conducted. Users are invited to listen to five groups of mashups per time, and each group has two mashups – generated by AMU and by our systems, in random order. 18 males and 6 females aged around 20~60 with similar background to the previous experiment participated in this test. The result of this test is given in Figure 7, and the corresponding p-values are also reported. The lead units generated by our system are commonly rated higher than those created by AMU, under a confidence level of 95%. This again verified the advantage of taking the harmonic change balance weighting into consideration. In fact, most of the GBUs are simple textured. So the lead units generated by AMU is more likely to pick simple textured lead units.

8. CONCLUSION AND FUTURE WORK

In this paper, a novel system is proposed to effectively create music mashups. There are two main contributions done in our system. First, both vertical and horizontal mashabilities are taken into consideration. Through this, our system can create a mashup with multiple background and lead track segments, which provides much higher flexibility in making mashups than the systems proposed in previous studies. Second, by taking the newly proposed vertical mashability measurement into account, user study shows that the improvement in user satisfaction is statistically significant. The subjective evaluations also show that the four concatenation cases we analyzed play a critical role in generating enjoyable mashups.

Many aspects of our system can be extended. First, in the vertical stage, we could alternatively match the unit based on the compatibility of pitch of lead unit and the chord of the background unit [26] instead of the chroma similarity between the lead and the background units directly. Second, sometimes we found that the lead units chosen by our system are too different from one another, so that the created mashups would sound very abrupt. To prevent this situation, we may restrict the chosen lead units to be with certain characteristics analyzed in advance, e.g. timbre, style, and emotion. Finally, we could further investigate the effect of overlapping the lead units and the chorus track units. Even more, the background units can also be separated into instrumental track units and drum track units, etc.. Toward the study about how to combine all kinds of units reasonably may provide true solutions to create music mashups in all conditions.

9. ACKNOWLEDGMENTS

The authors are grateful to all the participants (CMLab DSP group members) who helped evaluate the results.

10. REFERENCES

- [1] DJ Mix Generator. <http://www.djprince.no/site/DMG.aspx>.
- [2] Mixed in Key. Mashup. <http://mashup.mixedinkey.com/HowTo>.
- [3] Mixer. <http://ccmixter.org>.
- [4] Gilberto Bernardes. *Composing Music by Selection: Content-Based Algorithmic-Assisted Audio Composition*. Phd thesis, University of Porto, 2014.
- [5] C. Cannam. Rubber Band Audio Time Stretcher Library, 2012. <http://breakfastquay.com/rubberband/>.
- [6] Dave Cliff. *Hang the DJ : Automatic Sequencing and Seamless Mixing of Dance-Music Tracks*. Technical report, HP Labs, 2000.
- [7] Matthew Davies, Adam Stark, Fabien Gouyon, and Masataka Goto. Improvasher: A Real-Time Mashup System for Live Musical Input. In *Proc. NIME*, pages 541–544, 2014.
- [8] Matthew E P Davies, Philippe Hamel, Kazuyoshi Yoshii, and Masataka Goto. AutoMashUpper: An Automatic Multi-Song Mashup System. In *Proc. ISMIR*, Curitiba, PR, Brazil, 2013.
- [9] Matthew E. P. Davies, Philippe Hamel, Kazuyoshi Yoshii, and Masataka Goto. AutoMashUpper: Automatic Creation of Multi-Song Music Mashups. *IEEE Trans. ASLP*, 22(12):1726–1737, December 2014.
- [10] Steven B. Davis and Paul Mermelstein. Comparison of Parametric Representations for Monosyllabic Word Recognition in Continuously Spoken Sentences. *IEEE Trans. ASLP*, 28(4):357–366, 1980.
- [11] Simon Dixon. Evaluation of the Audio Beat Tracking System BeatRoot. *J. New Music Res.*, 36(1):39–50, 2007.
- [12] Mark Dolson. The Phase Vocoder: a Tutorial. *Computer Music Journal*, 10(4):14–27, 1986.
- [13] Garth Griffin, YE Kim, and Douglas Turnbull. Beat-Sync-Mash-Coder: a Web Application for Real-Time Creation of Beat-Synchronous Music Mashups. In *Proc. ICASSP*, pages 2–5, Dallas, Texas, USA, 2010.
- [14] Steven Hargreaves, Anssi Klapuri, and Mark Sandler. Structural Segmentation of Multitrack Audio. *IEEE Trans. ASLP*, 20(10):2637–2647, 2012.
- [15] Hiromi Ishizaki, Keiichiro Hoashi, and Yasuhiro Takishima. Full-Automatic DJ Mixing System with Optimal Tempo Adjustment based on Measurement Function of User Discomfort. In *Proc. of ISMIR*, pages 135–140, Kobe, Japan, 2009.
- [16] Tristan Jehan. *Creating Music by Listening*. Phd dissertation, Massachusetts Institute of Technology, 2005.
- [17] Rensis Likert. A Technique for the Measurement of Attitudes. *Archives of Psychology*, 22(140):1–55, 1932.
- [18] Heng-Yi Lin, Yin-Tzu Lin, Ming-Chun Tien, and Ja-Ling Wu. Music Paste: Concatenating Music Clips Based on Chroma and Rhythm Features. In *Proc. ISMIR*, Kobe, 2009.
- [19] Yin-Tzu Lin, Chuan-Lung Lee, Jyh-Shing Roger Jang, and Ja-Ling Wu. Bridging Music via Sound Effect Insertion. *IEEE Multimedia*. (to appear).
- [20] Yin-Tzu Lin, I-Ting Liu, Jyh-Shing Roger Jang, and Ja-Ling Wu. Audio Musical Dice Game: A User-preference-aware Medley Generating System. *ACM TOMM*, 11(4), 2015.
- [21] Matthias Mauch and Simon Dixon. Approximate Note Transcription for the Improved Identification of Difficult Chords. In *Proc. ISMIR*, pages 135–140, 2010.
- [22] Yizhao Ni, Matt McVicar, Paul Santos-Rodriguez, and Tjil De Bie. An End-to-End Machine Learning System for Harmonic Analysis of Music. *IEEE Trans. ASLP*, 20(6):1771–1783, 2012.
- [23] D. Robinson. *Perceptual Model for Assessment of Coded Audio*. Phd thesis, University of Essex, 2002.
- [24] Diemo Schwarz. *Data-driven concatenative sound synthesis*. Phd thesis, University of Paris 6 – Pierre et Marie Curie, 2004.
- [25] S. Siegel and N.J. Castellán. *Nonparametric statistics for the behavioral sciences*. McGraw-Hill, Inc., 1956.
- [26] Ian Simon, Dan Morris, and Sumit Basu. MySong: Automatic Accompaniment Generation for Vocal Melodies. In *Proc. SIGCHI*, pages 725–734, 2008.
- [27] Keita Tsuzuki, Tomoyasu Nakano, Masataka Goto, Takeshi Yamada, and Shoji Makino. Unisoner: An Interactive Interface for Derivative Chorus Creation from Various Singing Voices on the Web. In *Proc. ICMC*, number September, pages 790–797, 2014.
- [28] Stephen Webber. *DJ Skills: The Essential Guide to Mixing and Scratching*. Focal Press, 2007.
- [29] Melissa Hok Cee Wong. Mash-up. In Charles Hiroshi Garrett, editor, *The Grove Dictionary of American Music*. 2 edition, 2013.

HIERARCHICAL EVALUATION OF SEGMENT BOUNDARY DETECTION

Brian McFee^{1,2}, Oriol Nieto², and Juan P. Bello²

¹Center for Data Science, New York University

²Music and Audio Research Laboratory, New York University

^{1,2}{brian.mcfee, oriol, jpbello}@nyu.edu

ABSTRACT

Structure in music is traditionally analyzed hierarchically: large-scale sections can be sub-divided and refined down to the short melodic ideas at the motivic level. However, typical algorithmic approaches to structural annotation produce flat temporal partitions of a track, which are commonly evaluated against a similarly flat, human-produced annotation. Evaluating structure analysis as represented by flat annotations effectively discards all notions of structural depth in the evaluation. Although collections of hierarchical structure annotations have been recently published, no techniques yet exist to measure an algorithm’s accuracy against these rich structural annotations. In this work, we propose a method to evaluate structural boundary detection with hierarchical annotations. The proposed method transforms boundary detection into a ranking problem, and facilitates the comparison of both flat and hierarchical annotations. We demonstrate the behavior of the proposed method with various synthetic and real examples drawn from the SALAMI dataset.

1. INTRODUCTION

The analysis of structure in music is a principal area of interest to musicologists. Its goal is to identify and characterize the form of a musical piece by investigating the organization of its components, such as sections, phrases, melodies, or recurring motives. Traditional analyses usually provide multiple levels of annotation (*e.g.*, Schenkerian analysis), which suggest that music is structured hierarchically [3], and can be modeled and analyzed using tree representations [2].

In the music information research literature, *music segmentation* (also known as *music structure analysis*) is a task that aims to automatically identify the structure of a musical recording [6]. The segmentation task has historically been geared toward algorithms which produce a flat partition of the recording into disjoint segments. This formalization contrasts with our intuition that music exhibits hierarchical structure [7, 8]. Even though a large dataset of

hierarchically-structured human annotations is now publicly available [8], current evaluation methodologies are defined only for *flat* segmentations. As a result, the dimension of *depth* has been practically ignored in the evaluation of music segmentation algorithms.

In contrast to segmentation, the *pattern discovery* task formulation allows output segments to overlap, and the annotation is not required to cover the entire piece. These two tasks share multiple attributes [5], and steps toward a general formulation musical structure analysis could be made by accounting for depth in segmentation. Numerous metrics to evaluate pattern discovery have been proposed [1]. However, they are designed to capture repeated patterns, and would be inappropriate for evaluating non-repeating, hierarchical structure.

1.1 Our contributions

We present the *Tree Measures* (*T*-measures): an evaluation framework designed to measure the accuracy of boundary detection in hierarchical segmentations. The *T*-measures infer frame-wise similarity from a hierarchical annotation, and then compare the induced rank-orderings to assess agreement between reference and estimated annotations. The *T*-measures integrate information from all layers of a hierarchy, trivially specialize to handle flat annotations, and require no explicit correspondence between the depth of the estimated and reference hierarchies. Thus, the *T*-measures encourage the development of new algorithms to produce richer representations of structure. Although not all music can necessarily be modeled using trees [11], we argue that tree-based evaluation represents a first step toward moving beyond flat structure analyses. We demonstrate the properties of *T*-measures with multiple synthetic, human, and algorithmic examples.

2. SEGMENT BOUNDARY EVALUATION

Segmentation algorithms are typically evaluated for two distinct goals. The first goal, *boundary detection*, evaluates the algorithm’s ability to detect the times of transitions between segments. The second goal, *structural grouping*, evaluates the labeling applied to the estimated segmentation, and thus quantifies the ability of an algorithm to detect repeated forms, such as verses or refrains. In this paper, we focus exclusively on the boundary detection task.

Boundary estimates are typically evaluated by precision and recall [10]. Estimated and reference boundaries are



© Brian McFee, Oriol Nieto, Juan P. Bello.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Brian McFee, Oriol Nieto, Juan P. Bello. “Hierarchical Evaluation of Segment Boundary Detection”, 16th International Society for Music Information Retrieval Conference, 2015.

matched within a specified tolerance window — typically either 0.5 or 3 seconds — and the hit rate n_h (number of matches) is used to define precision and recall scores:

$$P := \frac{n_h}{n_e}, \quad R := \frac{n_h}{n_r}, \quad (1)$$

where n_e and n_r denote the number of boundaries in the estimated and reference annotations, respectively. P and R are typically combined into a single F -measure by computing their harmonic mean.

Boundary detection has also been evaluated by *deviation* [10]. This is done by measuring the median time (absolute) differential between each reference boundary and the nearest estimated boundary ($R2E$), and vice versa ($E2R$). Boundary deviation is useful for quantifying the temporal accuracy of a detection event. However, it can be sensitive to the number of estimated boundaries.

2.1 The limitations of flat evaluation

The precision-recall paradigm has been critical to quantifying improvements in segmentation algorithms, but it has numerous limitations with hierarchical annotations. The most obvious limitation is that both the reference and estimated annotations must have flat structure. This is sometimes resolved by collecting multiple flat reference annotations for each track, each corresponding to different levels of analysis [8].

When only the estimation is flat, it is still not obvious how to compute accuracy against multiple layers. Aggregating reference boundaries across layers prior to evaluation would imply that all boundaries are equally informative. However, high-level boundaries often convey more information about the overall structure of the piece, but their contribution to the total score may be diluted by the abundance of low-level boundaries, which necessarily outnumber high-level boundaries in hierarchical annotations.

Flat evaluation followed by aggregation across layers can be similarly problematic, since it discards the relational structure between layers in the reference annotation. This can complicate interpretation of the scores by conflating inaccurate boundary detection with mismatch between the target levels of the estimate and reference annotations [9].

Finally, the above strategies provide no means to directly compare two hierarchical annotations. While one may imagine simple comparison strategies when both hierarchies have a small number of layers with an obvious layer-wise correspondence — e.g., SALAMI's *large-* and *small-scale* annotations — it is unclear how to proceed in more general settings.

3. THE TREE MEASURES

In this section, we derive the *tree measures* for evaluating multi-level segment boundary detection. The evaluation is based on a reduction to ranking evaluation, which we describe in detail below.

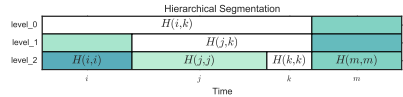


Figure 1: An example of a three-level hierarchical segmentation. Frames i , j , and k are indicated along the x -axis, and their containing segments are indicated within the figure, e.g., $H(j, k)$.

3.1 Preliminaries

Let X denote a set of sample frames generated from the track at some fixed resolution f_r (e.g., 10Hz).¹ Let S denote a flat, temporally contiguous partition of X , and let $S(i)$ identify the segment containing the i th frame in X . We will use the subscripts S_R and S_E to denote *reference* and *estimated* annotations, respectively.

A *hierarchical segmentation* H is defined as a tree of flat segmentations (S^0, S^1, \dots, S^d) where each layer is a *refinement* of the preceding layer.² Let $H(i, j)$ identify the smallest (most refined) segment containing frames i and j . We will denote precedence (containment) of segments by \prec : e.g., $H(j, k) \prec H(i, k)$. Note that flat segmentations are a special case of hierarchical segmentations, where there are only two levels of segmentation, and the first layer contains no boundaries.

As illustrated in Figure 1, hierarchical segmentations can be represented as tree structures. Here, $H(i, i)$, $H(j, j)$ and $H(k, k)$ denote the most specific segments containing frame i , j and k , respectively. From the figure, we observe that $H(j, k)$ identifies the least common ancestor of frames j and k . We can generally infer membership and precedence relations from the hierarchy, e.g.,

$$j \in H(j, j) \prec H(j, k) \prec H(i, j) = H(i, k). \quad (2)$$

3.2 Flat segmentation and bipartite ranking

Segmentation evaluation can be reduced to a ranking evaluation problem as follows. Let q denote an arbitrary frame, and let i and j denote any two frames such that $S_R(q) = S_R(i)$ and $S_R(q) \neq S_R(j)$. In this case, i may be considered *relevant* for q , and j is considered *irrelevant*. This leads to the following per-frame recall metric:

$$f(q; S_E, S_R) := \frac{\sum_{\substack{i \in S_R(q) \setminus \{q\} \\ j \notin S_R(q)}} \mathbb{I}[S_E(q) = S_E(i) \neq S_E(j)]}{Z_q} \quad (3)$$

$$Z_q := (|S_R(q)| - 1) \cdot (n - |S_R(q)| + 1),$$

where $\mathbb{I}[\cdot]$ is the indicator function, $n = |X|$ denotes the total number of frames, and Z_q counts the number of terms in the summation. The score for frame q is the fraction of

¹ Non-uniform samplings (e.g., beat- or onset-aligned samples) are also easily accommodated.

² A partition S^{i+1} is a refinement of partition S^i if each member of S^{i+1} is contained within exactly one member of S^i .

pairs (i, j) for which S_E agrees with S_R with respect to q . Averaging over all q yields a mean recall score:

$$\rho(S_E, S_R) := \frac{1}{n} \sum_q f(q; S_E, S_R). \quad (4)$$

3.3 Hierarchies and partial ranking

Equation (3) is defined in terms of segment membership equality, but it has a straightforward generalization to hierarchical segmentations. If we restrict attention to a query sample q , then $H(q, \cdot)$ induces a partial ranking over the remaining samples. Frames contained in $H(q, q)$ are considered maximally relevant, followed by those in $H(q, q)$'s immediate ancestor, and so on.

Rather than compare frames q , i , and j where $S(q) = S(i) \neq S(j)$, we can instead compare where $H(q, i) \prec H(q, j)$: *i.e.*, the pair (q, i) merge deeper in the hierarchy than do (q, j) . This leads to the following generalization of Equation (3):

$$g(q; H_E, H_R) := \sum_{\substack{(i,j), \\ i \neq q, \\ H_R(q,i) \prec H_R(q,j)}} \frac{[H_E(q,i) \prec H_E(q,j)]}{Z_q}, \quad (5)$$

where Z_q is suitably modified to count the number of terms in the summation. This definition is equivalent to Equation (3) for flat hierarchies, but it applies more generally to hierarchies of arbitrary (and unequal) depth.

Just as in Equation (3), g can be viewed as a classification accuracy of correctly predicting pairs (i, j) as positive (q and i merge first) or negative (q and j merge first). Ties ($H(q, i) = H(q, j)$) are precluded by the strict precedence operator in the summation. Equation (5) can be alternately be viewed as a generalized area under the curve (AUC) over the partial ranking induced by the hierarchical segmentation, where depth within the estimated hierarchy H_E plays the role of the detection threshold.

Averaging over q yields the *tree-recall* T -measure:

$$\mathcal{T}_R(H_E, H_R) := \frac{1}{n} \sum_q g(q; H_E, H_R). \quad (6)$$

The *tree-precision* metric $\mathcal{T}_P(H_E)$ is defined analogously by swapping the roles of H_E and H_R :

$$\mathcal{T}_P(H_E, H_R) := \mathcal{T}_R(H_R, H_E). \quad (7)$$

Intuitively, \mathcal{T}_R measures how many triplets generated by the reference H_R can be found in the estimate H_E , while \mathcal{T}_P computes the converse. The T -measures retain interpretation as recall and precision scores, albeit at the level of frame triplets rather than boundaries. Finally, an analogous F -measure \mathcal{T}_F can be defined in the usual way by computing the harmonic mean of \mathcal{T}_P and \mathcal{T}_R .

3.4 Windowing in Time

The T -measures defined above capture the basic notion of hierarchically nested, frame-level relevance, but they pose three technical limitations. First, the score for each query

will generally depend on the track duration n , which makes comparisons between tracks of differing length problematic. Second, for large values of n (long tracks), Equation (5) can be dominated by trivial comparisons where j lies far from q in time, *i.e.*, $|q - i| \ll |q - j|$. Longer tracks will produce inflated scores compared to shorter tracks, simply by having more “easy” comparisons. Finally, the calculation of Equation (6) can be expensive, taking $\mathcal{O}(n^3)$ time using a direct implementation.

To resolve these issues, we introduce a time window of w seconds to both simplify the calculation of the metric and normalize its range. This is achieved by restricting the triples (q, i, j) in the summation such that i and j both lie within a window of w seconds centered at q . Adding this windowing property to equations (5, 6) yields the windowed T -measures:

$$g(q; H_E, H_R, w) := \sum_{\substack{i,j \in \{x: |q-x| \leq w/2\} \\ i \neq q, \\ H_R(q,i) \prec H_R(q,j)}} \frac{[H_E(q,i) \prec H_E(q,j)]}{Z_q(w)}, \quad (8)$$

$$\mathcal{T}_R(H_E, H_R; w) := \frac{1}{n} \sum_q g(q; H_E, H_R, w), \quad (9)$$

and $Z_q(w)$ is again modified to count the terms in the summation. This reduces computational complexity from $\mathcal{O}(n^3)$ to $\mathcal{O}(nw^2)$. Each query frame q now operates over a bounded number of comparisons, so the windowed T -measures are calibrated across tracks of different lengths. This property is useful when compiling score statistics over a test collection.

3.5 Transitive reduction

Just as Equation (5) can be dominated by long-range interactions in the absence of windowing, deep hierarchies can also pose a problem. To see this, consider the sequence $H_R(q, i) \prec H_R(q, j) \prec H_R(q, k)$. Since the summation in Equation (5) ranges over all precedence comparisons, and $i \in H_R(q, j)$, the triple (q, i, k) is double-counted. Since segments grow in size at higher levels in the hierarchy, over-counting can dominate the evaluation.

To counteract this effect, the summation can be restricted to include only direct precedence relations. This is accomplished by comparing samples only from successive levels in the hierarchy, *i.e.*, replacing the partial ranking generated by q with its transitive reduction. This both eliminates redundant comparisons and increases g 's effective range. We refer to the resulting metrics as *reduced* T -measures.

4. SYNTHETIC EXAMPLES

In this section we discuss the behavior of the T -measures by showing various synthetic examples, and comparing them against other existing methods when possible. For each example in this section, we illustrate the behavior of our proposed metric under different window times w . This section is subdivided by the types of annotations under consideration.

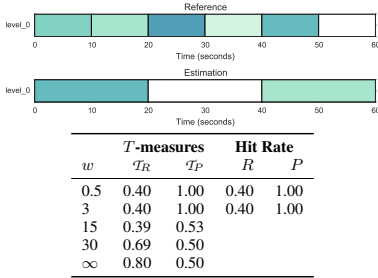


Figure 2: Flat vs. flat boundaries (top), T -measures and boundary detection (hit-rate) scores (bottom).

4.1 Flat vs. flat annotations

We first compare two flat boundary annotations to demonstrate how the T -measures behave compared to standard boundary detection. When both annotations are flat, the reduced T -measures behave identically to the full measures, so we omit them from this section. The synthesized flat boundaries are displayed on the top of Figure 2, and they aim to capture a situation where an algorithm correctly detects a subset of the reference boundaries.

The hit rate scores obtain a recall of 0.40 and a precision of 1.0, since all estimated boundaries are also in the reference, but only two out of five boundaries were retrieved.³ When w does not exceed the minimum segment duration, the T -measures coincide exactly with the boundary detection metrics. For larger w , \mathcal{T}_P decreases, while \mathcal{T}_R increases as w approaches the track duration. The dependency on w is further explored in Section 5.1.

To understand the relationship between \mathcal{T}_P and w , consider the example $(q, i, j) = (5, 15, 25)$. The estimation considers i to be relevant for q (since they belong to the segment $[0, 20]$), and j to be irrelevant for q . Meanwhile, the reference considers both i and j to be equally irrelevant for q , so this triple contributes 0 to the precision metric. Note that this comparison is counted only when w is large enough to span multiple segments.

In general, sensitivity to long-range interactions increases with w . This illustrates how the window size depends on the duration and scale of structure that the practitioner wishes to capture.

4.2 Flat vs. hierarchical annotations

Here we present four examples of flat estimations against a fixed hierarchical reference, but note that the reverse comparisons can be inferred by swapping \mathcal{T}_P and \mathcal{T}_R .

4.2.1 Large-scale and under-segmentation

Figure 3 illustrates a flat estimation corresponding to the highest layer of a hierarchical reference. We report T -

³ The first and last boundaries (0 and 60s) mark the beginning and end of the track, and since they are constant across all estimates, we suppress them during the evaluation to avoid score inflation.

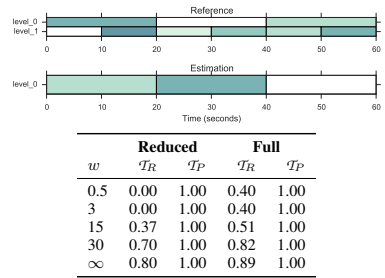


Figure 3: Hierarchical reference vs. flat (large-scale) estimation (top) and T -measures (bottom). *Reduced* uses the transitive reduction method of section 3.5, while *Full* uses comparisons across all layers.

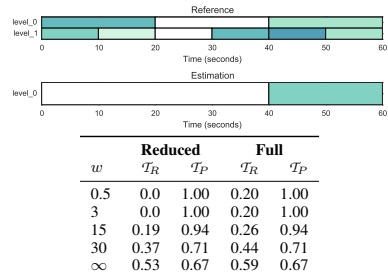


Figure 4: Hierarchical reference vs. flat under-segmentation (top) and T -measures (bottom).

measures with and without the transitive reduction strategy described in Section 3.5. The T -measures behave as expected: the tree-precision score \mathcal{T}_P is always 100%, since the reference contains the estimation. We also observe the general trend that *full* scores exceed *reduced* scores.

For small time windows ($w \leq 3$), the full tree-recall score is 40%, just as in the previous example. The *reduced* recall scores in this case are 0 because no frame q in the estimation has two frames i, j both within $w \leq 3$ seconds that merge within one layer of each-other in the reference.

Figure 4 illustrates an example of under-segmentation: the estimation misses a high-level structural change at 20s. Again, small w yields T -measures which coincide with standard boundary detection metrics. Larger w increases the tree-recall (and decreases precision) since only long-range interactions are well represented in the estimation.

4.2.2 Small-scale and over-segmentation

Figure 5 illustrates an example comparable to Figure 3, except that the estimation now corresponds to the bottom layer of the reference annotation. Again, since the reference contains the estimation, precision is maximal for all w . However, the reference provides strictly more informa-

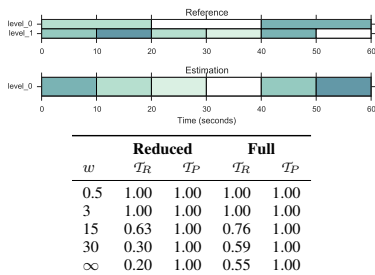


Figure 5: Hierarchical reference vs. flat, small-scale estimation (top) and T -measures (bottom).

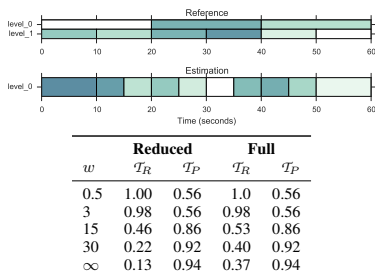


Figure 6: Hierarchical reference vs. flat over-segmentation (top) and T -measures (bottom).

tion: namely, it encodes structure over the low-level segments. The T -measures quantify the missing information in the estimation. When w exceeds the smallest segment duration (10s), \mathcal{T}_R decreases. This information would be obscured by independent, layer-wise boundary evaluation.

Similarly, Figure 6 illustrates an *over-segmentation* where the estimation predicts more boundaries than the deepest layer of the reference. Again, the \mathcal{T}_R decays when the window captures multiple short segments. Unlike the under-segmented example in Figure 4, long-range interactions derived from H_E are mostly satisfied by H_R , so \mathcal{T}_P increases rather than decreases.

4.3 Hierarchical vs. hierarchical

Figure 7 compares two different hierarchical segmentations. The estimation contains an additional high-level layer, but is otherwise identical to the reference. At small w , both T -measures agree perfectly, since the window is not large enough to resolve differences. As w increases, \mathcal{T}_P decreases as expected, since the estimation has found an additional structural element not captured in the reference. The \mathcal{T}_R scores remain at 100% for all w .

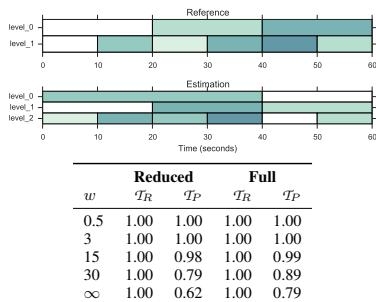


Figure 7: 2-layer vs. 3-layer hierarchical boundaries (top) and T -measures scores (bottom).

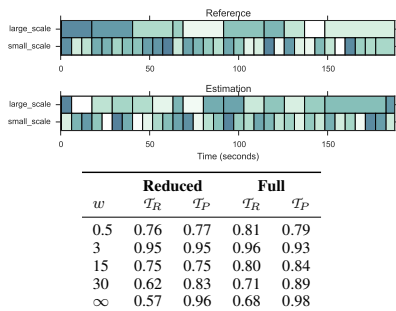


Figure 8: Hierarchical annotations for SALAMI track #636 from the two different human annotators. Top: annotations; bottom: T -measures scores.

5. LARGE-SCALE EVALUATION

In this section, we apply the T -measures to quantify inter-annotator agreement in the SALAMI corpus, and evaluate the hierarchical predictions of the agglomerative clustering method (OLDA) of McFee and Ellis [4].

5.1 Human annotator agreement

Figure 8 illustrates hierarchical annotations obtained from two human annotators on one track in the SALAMI dataset. While the two annotators tend to agree at the small scale, they differ at the large scale. This is reflected in the T -measures: at large w , the recall skews low because the reference's large-scale annotations are coarser than those of the estimation.

To further investigate inter-annotator agreement, we computed T -measure scores between hierarchical reference annotations for the 410 tracks in the SALAMI dataset where two annotations are available and both mark the start and end times of the song equally at both levels. To simplify exposition, we summarize agreement by \mathcal{T}_P . Figure 9

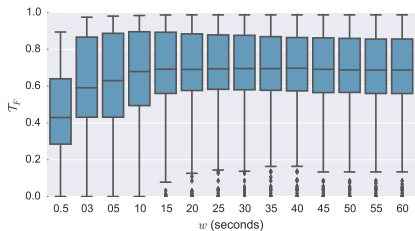


Figure 9: \mathcal{T}_F scores between human annotators for SALAMI tracks over a range of window sizes w .

illustrates the distribution of per-track \mathcal{T}_F scores as a function of w . We observe that the score distribution is relatively stable for $w \geq 15$.⁴ The example in Figure 8 is generally representative of inter-annotator agreement, achieving $\mathcal{T}_F = 0.75$ at $w = 15$. The out-lying low scores tend to be examples where one annotator ignored structure annotated by the other: e.g., in track #68, one annotator only marked *silence* boundaries.

This analysis quantitatively substantiates prior observations that humans do not perfectly agree upon structural annotations [9], and suggests an accuracy ceiling near 70% for hierarchical annotation. Similarly, it suggests that $w = 15$ provides a reasonable default value for the SALAMI dataset. This setting is large enough to capture multiple small-scale segments: in the tracks considered for this evaluation, the median small-scale segment duration was 6.66s, with a 95th percentile of 15.69s.

5.2 Annotator vs. algorithm

Finally, we evaluated the quality of hierarchical segmentations produced by OLDA [4].⁵ Figure 10 illustrates one example output of OLDA and the resulting T -measures. The reference provides two levels of segmentation (large and small), while the estimation produces several layers with generally large segments. For sufficiently large w , the estimation achieves high recall and low precision. This behavior is typical of the OLDA method, which constructs hierarchies in a bottom-up fashion by agglomerative clustering, adding only a single boundary at each layer. Due to the depth of the estimated boundaries, the *full* scores are inflated compared to the *reduced* scores.

Figure 11 displays the \mathcal{T}_F score distribution for OLDA, measured against annotator 1 on 726 tracks from SALAMI. These results reveal a gap of around 30% between inter-annotator agreement (Figure 9) and the performance of OLDA. This suggests that there is substantial room for improvement in hierarchical boundary estimation algorithms.

⁴ The analogous plots for \mathcal{T}_P and \mathcal{T}_R are omitted for brevity, but illustrate the same trend.

⁵ To the authors' knowledge, this is the only published method for hierarchical boundary detection.

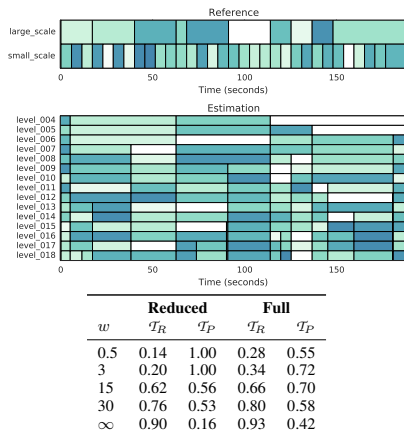


Figure 10: Hierarchical reference annotation vs. OLDA on SALAMI track #636. (top) and T -measures (bottom).

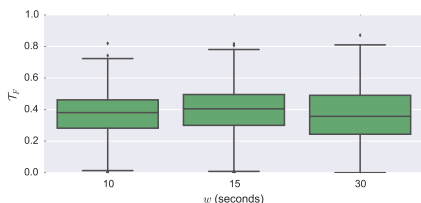


Figure 11: \mathcal{T}_F scores between OLDA and human reference annotations on the SALAMI dataset.

6. DISCUSSION AND CONCLUSIONS

The implementation of T -measures depends upon two critical parameters: the time window w , and whether to use the *reduced* or *full* metrics. While the setting of w ultimately depends upon the practitioner's preference and characteristics of the dataset, the results on SALAMI suggest that $w = 15$ provides a reasonable balance between capturing high-level structure and resilience to long-range interactions. As illustrated in section 4.2.1, when w is large enough to capture multiple short segments, the transitive reduction approach can also be used to enhance the range of the metrics while eliminating redundant comparisons.

In this paper, we focused only on the problem of evaluating estimated boundaries. In future work, we plan to extend general ideas behind T -measures to other structural annotation problems, such as segment label agreement.

7. ACKNOWLEDGEMENTS

BM acknowledges support from the Moore-Sloan Data Science Environment at NYU.

8. REFERENCES

- [1] Tom Collins. *Discovery of Repeated Themes & Sections*, 2013.
- [2] Fred Lerdahl and Ray Jackendoff. *A Generative Theory of Tonal Music*. MIT Press, 1983.
- [3] Fred Lerdahl and Ray Jackendoff. An Overview of Hierarchical Structure in Music. *Music Perception: An Interdisciplinary Journal*, 1(2):229–252, 1983.
- [4] Brian McFee and Daniel P. W. Ellis. Learning to Segment Songs with Ordinal Linear Discriminant Analysis. In *Proc. of the 39th IEEE International Conference on Acoustics Speech and Signal Processing*, Florence, Italy, 2014.
- [5] Oriol Nieto and Morwaread M. Farbood. Identifying Polyphonic Patterns From Audio Recordings Using Music Segmentation Techniques. In *Proc. of the 15th International Society for Music Information Retrieval Conference*, pages 411–416, Taipei, Taiwan, 2014.
- [6] Jouni Paulus, Meinard Müller, and Anssi Klapuri. Audio-Based Music Structure Analysis. In *Proc of the 11th International Society of Music Information Retrieval*, pages 625–636, Utrecht, Netherlands, 2010.
- [7] Geoffroy Peeters and Emmanuel Deruty. Is Music Structure Annotation Multi-Dimensional? A Proposal for Robust Local Music Annotation . In *Proc. of the 3rd International Workshop on Learning Semantics of Audio Signals*, pages 75–90, Graz, Austria, 2009.
- [8] Jordan B. Smith, J. Ashley Burgoyne, Ichiro Fujinaga, David De Roure, and J. Stephen Downie. Design and Creation of a Large-Scale Database of Structural Annotations. In *Proc. of the 12th International Society of Music Information Retrieval*, pages 555–560, Miami, FL, USA, 2011.
- [9] Jordan B. L. Smith and Elaine Chew. A Meta-Analysis of the MIREX Structure Segmentation Task. In *Proc. of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil, 2013.
- [10] Douglas Turnbull, Gert RG Lanckriet, Elias Pampalk, and Masataka Goto. A supervised approach for detecting boundaries in music using difference features and boosting. In *ISMIR*, pages 51–54, 2007.
- [11] Geraint A. Wiggins and Jamie Forth. Idyot: A computational theory of creativity as everyday reasoning from learned information. In Tarek R. Besold, Marco Schorlemmer, and Alan Smaill, editors, *Computational Creativity Research: Towards Creative Machines*, volume 7 of *Atlantis Thinking Machines*, pages 127–148. Atlantis Press, 2015.

IMPROVING MIDI GUITAR’S ACCURACY WITH NMF AND NEURAL NET

Masaki Otsuka and Tetsuro Kitahara

Graduate School of Integrated Basic Sciences, Nihon University

{masaki, kitahara}@kthrlab.jp

ABSTRACT

In this paper, we propose a method for improving the accuracy of MIDI guitars. MIDI guitars are useful tools for various purposes from inputting MIDI data to enjoying a jam session system, but existing MIDI guitars do not have sufficient accuracy in converting the performance to an MIDI form. In this paper, we make an attempt on improving the accuracy of a MIDI guitar by integrating it with an audio transcription method based on non-negative matrix factorization (NMF). First, we investigate an NMF-based algorithm for transcribing guitar performances. Although the NMF is a promising method, an effective post-process (i.e., converting the NMF’s output to an MIDI form) is a non-trivial problem. We propose use of a neural network for this conversion. Next, we investigate a method for integrating the outputs of the MIDI guitar and NMF. Because they have different tendencies in wrong outputs, we take an policy of outputting only common parts in the two outputs. Experimental results showed that the F-score of our method was 0.626 whereas those of the MIDI-guitar-only and NMF-and-neural-network-only methods were 0.347 and 0.526, respectively.

1. INTRODUCTION

A MIDI guitar, which outputs the user’s performance data into the MIDI format in real time, is useful for guitarists to engage in various music activities such as inputting MIDI data into a computer and enjoying the use of a jam session system. However, the accuracy of MIDI guitars is not as high as a MIDI keyboard because the MIDI guitar detects the strings’ vibration by analyzing the temporal changes in the magnetic field around the strings.

There have been many attempts made to transcribe guitar performances [1–3, 5, 8–10, 12]. Arimoto et al. remade the PreFEst method, originally developed by Goto [4], for a guitar based on physical constraints on fingering forms [1]. Yazawa et al. also focused on latent harmonic allocation for a guitar based on physical constraints in fingering form [12]. Barbancho et al. furthermore investi-


gated these physical constraints [2]. Fiss et al. constructed a system that transcribes a guitar performance as a tablature [3]. This system estimates not only the notes that are played, but it also estimates how the notes are played (i.e., string number and fret number) through audio signal processing. O’Grady et al. considered both the use of non-negative matrix factorization (NMF) [7] and a hardware improvement of a MIDI guitar for accurate guitar performance transcription [8]. Harquist also proposed a real-time guitar transcription method using NMF [5]. In addition, there have been attempts to improve guitar performance transcription by integrating audio signal processing with computer vision [9, 10].

In this paper, we focus on improving the accuracy of MIDI guitars by integrating them with audio signal processing technologies (especially NMF). Almost all MIDI guitars have an audio output jack for connecting to a guitar amplifier as well as a MIDI output. By connecting this audio output jack to a PC’s audio input jack, the guitar’s audio signal can be analyzed. By inputting the guitar’s MIDI output to that PC, the audio result and the guitar’s MIDI output can be integrated. Thus, introducing audio signal processing to a MIDI guitar does not require any special equipment or hardware improvements. O’Grady et al. focused on a similar technique as we employ here but their method involved hardware improvements [8]; our methodology requires no hardware improvement. Using computer vision [9, 10] is an interesting approach, but it requires installing a camera and it may restrict the player’s motions. Using physical constraints in fingering forms [1, 2, 12] is a common and promising approach, but we dare to adopt the approach of exploring how much we can improve the accuracy without using physical constraints. Physical constraints can be applied to our method for further improvements in the future.

The rest of this paper is organized as follows: In Section 2 we propose a method for transcribing guitar performance using NMF. In Section 3 we describe a method for integrating NMF-based transcription outputs and the MIDI guitar’s outputs. In Section 4 we report our experimental results. Finally, we conclude the paper in Section 5.

2. AUDIO-TO-MIDI CONVERSION WITH NMF

NMF is a technique for decomposing a matrix V into the product of two matrices W and H , that is, $V \cong WH$, where W is a basis matrix and H is a gain matrix. A typical

 © Masaki Otsuka and Tetsuro Kitahara. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Masaki Otsuka and Tetsuro Kitahara. “Improving MIDI Guitar’s Accuracy with NMF and Neural Net”, 16th International Society for Music Information Retrieval Conference, 2015.

usage of NMF in automatic music transcription is to apply NMF to a spectrogram. Then, the basis matrix W is an array of N column vectors w_n that represent the spectrum of each note n ; the gain matrix is an array of N row vectors h_n that represent a temporal sequence of the gain for the basis vector w_n . Because the gain vector h_n represents an approximation of a temporal sequence of the amplitude for the note n , the onset and offset times for a note n can be identified by thresholding h_n ; steep rises in the time series $\{h_{n,t}\}_t$ represent onsets and steep drops represent offsets.

However, there are two problems with this technique. The first one is that standard NMF is applicable only after the entire spectrogram is obtained. This fact means that standard NMF cannot be used for real-time processing. The second problem is that it is difficult to determine a universally appropriate threshold because the actual gains vary according to playing style, strings, and other factors. Thus, the issues to be resolved here can be summarized as follows:

Issue 1 How to apply NMF to real-time processing

Issue 2 How to determine an appropriate threshold depending on the playing style, strings, etc.

In this paper, we resolve these issues as follows:

Solution 1 We ask the user to play a chromatic scale (from the lowest note to the highest note) for each string in advance and apply NMF to this *preliminary performance*. We assume that the spectrum of each note is similar enough between the preliminary and target performances¹ if the same person plays the same instrument in the same way. Under this assumption, the basis matrix calculated from the preliminary performance is then used to obtain the gain vectors for the target performance.

Solution 2 We introduce one more preliminary performance and adaptively determine the threshold. This preliminary performance has a similar musical feature to the target performance, and we ask the user to play the phrase specified by the system accurately (thus, the system knows the ground truth). Adaptation of the threshold using these data is approximately equivalent to learning a neural network. We therefore learn how high the gain is and how steeply the gain rises at onsets with a neural network and use this neural network for detecting onsets.

In the rest of this section, we first describe a method in which only Solution 1 is introduced (we call this method the *baseline method*). Next, we introduce Solution 2 to this baseline method.

¹ The *target performance* refers to the performance to be converted to the MIDI format.

2.1 Baseline method — Introducing Solution 1 only

Stage 1: Estimating basis matrix from preliminary performance

After the user plays all chromatic notes successively for each string k (called the *1st preliminary performance*), the spectrogram V_k is calculated using the short-term Fourier transform with a 4096-point Hamming window shifted by 10 ms (we suppose 44.1-kHz sampling). Then, the spectrogram V_k is decomposed into the basis matrix W_k and the gain matrix H_k using NMF. To avoid that spectral peaks for different notes are mixed into a single basis vector, we prepare 35 basis vectors for each string even though each string has 23 notes. We then obtain 23 basis vectors by merging pairs of basis vectors that have a high cosine similarity.

Stage 2-1: Estimating gain vectors for target performance

The user plays the target performance (i.e., the performance to be converted to the MIDI format). As the user plays, the power spectrum v_t (where t is time) is obtained via the Fourier transform, and then the gain vector $h_{t,k}$ for each string k is calculated. The gain vector $h_{t,k}$ is defined as $h_{t,k} = W_k^{-1}v_t$, where W_k is the basis matrix for the string k , obtained in Stage 1. Because W_k is not a square matrix in general, its inverse matrix cannot be calculated. We therefore use a pseudo-inverse matrix [6] instead.

Stage 2-2: Generating MIDI messages by thresholding gain vectors

After the gain vector $h_{t,k}$ is calculated, MIDI messages are generated. When the n -th element of $h_{t,k}$ has a higher value than the threshold h_0 but that of $h_{t-1,k}$ does not (that is, $h_{t-1,k,n} \leq h_0 < h_{t,k,n}$), a MIDI Note On message for the note number corresponding to fret n of string k is generated. When $h_{t-1,k,n} > h_0 \geq h_{t,k,n}$, a MIDI Note Off message is generated. When $h_{t-2,k,n} > h_{t-1,k,n}$ and $h_{t-1,k,n} < h_{t,k,n}$, even if both $h_{t-1,k,n}$ and $h_{t,k,n}$ are higher than h_0 , we can consider that a note is played again before the previous note is decayed enough. If this is the case, a Note Off message is generated at time $t-1$ and a Note On message at time t .

2.2 Introducing Solution 2

The method discussed above involves thresholding but the appropriate threshold depends on various factors including the individual instrument, the strength of picking, and the characteristics of the player. In practice, dynamic adjustment of the threshold is not straightforward. We therefore add one more preliminary performance (called the *2nd preliminary performance*) and adjust the threshold using this 2nd preliminary performance under the assumption that a correct transcription of the 2nd preliminary performance has been given. Let $h_{t,k,n}$ be the gain in the 2nd preliminary performance at time t , string k , and fret n . Whether t is an onset time at fret n of string k in this performance

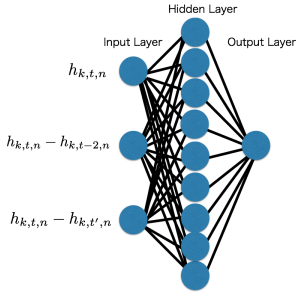


Figure 1. Neural network that we employ

can be identified from the correct transcription, then it is represented as follows:

$$s_{t,k,n} = \begin{cases} 1 & (t \text{ is an onset time at fret } n \text{ of string } k) \\ 0 & (\text{else}) \end{cases}$$

What to be solved here is to find h_0 such that $h_{t,k,n} > h_0$ iff $s_{t,k,n} = 1$. This h_0 can be estimated by minimizing $E(h_0) = \sum_{t,k,n} \{\zeta(-h_0 + h_{t,k,n}) - s_{t,k,n}\}^2$, where $\zeta(x)$ is the sigmoid function, that is, $\zeta(x) = 1/(1 + e^{-x})$. It is equivalent to training a neural network. The temporal differential of $h_{t,k,n}$ is also considered important for onset detection, so we obtain a neural network shown in Figure 1 by adding such features.

Stage 1: Estimating basis matrix from 1st preliminary performance

In the same way as Stage 1 of the baseline method, the 1st preliminary performance is played by the user, and then the basis matrix W_k for each string k is calculated.

Stage 2-1: Estimating gain vectors for 2nd preliminary performance

The user plays the 2nd preliminary performance. As he/she plays, the power spectrum v_t and the gain vector $h_{t,k} = W_k^{-1}v_t$ for each string k are calculated every 10 ms in the same way as in Stage 2-1 of the baseline method.

Stage 2-2: Learning neural network

For each element $h_{t,k,n}$ of $h_{t,k}$, the following steps are performed if $h_{t,k,n}$ is a peak:

1. Features are extracted from $h_{t,k,n}$ and are set to the vector $x_{t,k,n}$. In the current implementation, the following feature vector is used:

$$x_{t,k,n} = (h_{t,k,n}, h_{t,k,n} - h_{t-2,k,n}, h_{t,k,n} - h_{t',k,n}),$$

where t' is the time of the last valley before t in $\{h_{\tau,k,n}\}_{\tau=0,\dots,t}$, in other words, the maximum value of $\tau (< t)$ such that $h_{\tau,k,n} < h_{\tau-1,k,n}$ and $h_{\tau,k,n} < h_{\tau+1,k,n}$.

2. The supervision $s_{t,k,n}$ is defined as described above.

3. The neural networks shown in Figure 1 are trained using backpropagation such that the difference between the value of the output node $y_{t,k,n}$ and the supervision $s_{t,k,n}$ is minimized. We prepare and train different neural networks for different string, but we use the same neural network for different frets of the same string due to a limited number of training data. Each neural network has a single hidden layer consisting of three to ten nodes (we try all cases and present the best result).

In the training, the number of data with supervisions of 1 and 0 are balanced.

Stage 3-1: Estimating gain vectors for target performance

During the target performance, the spectrum and the gain vector are calculated every 10 ms in the same way as in Stage 2-1.

Stage 3-2: Generating MIDI messages based on neural network

The feature vector $x_{t,k,n}$ is calculated in the same way as in Stage 2-2. Then, the value of the output node $y_{t,k,n}$ in the trained neural network is calculated for each time, each string, and each fret. When this value is higher than 0.5, a MIDI Note On message for the corresponding note number is generated.

Theoretically, offsets can also be learned and estimated with a neural network. However, for simplicity, offsets are detected in the same way as in the baseline method.

3. INTEGRATION OF MIDI GUITAR AND NMF

In this section, we describe a method for integrating the outputs of a MIDI guitar and the method discussed in Section 2.2. When we discuss how to integrate two different outputs, we should consider a tradeoff between recall rates and precision rates. We believe that precision is more important in our task because false positives (MIDI messages generated but actually not played) directly result in dissonant sound; false negatives (MIDI messages not generated but actually played) do not. We therefore adopt an approach of outputting the common part of the two outputs.

Stages 1 to 3-2

We perform the same process as in Section 2.2 is performed until Stage 3-2. Although in the method in Section 2.2 the value of the output node $y_{t,k,n}$ is thresholded, it is not thresholded here because $y_{t,k,n}$ is used in Stage 4.

Stage 4: Integration with MIDI guitar outputs

From the output of the MIDI guitar, we obtain the following value:

$$m_{t,k,n} = \begin{cases} 1 - \alpha & \text{(the note corresponding to fret} \\ & \textit{n} \text{ of string } k \text{ is being played.)} \\ \alpha & \text{(else)} \end{cases}$$

“(A note is) being played” means the state in which the MIDI guitar had output a MIDI Note On message for this note number but has not yet output a MIDI Note Off message. Note that it represents the MIDI guitar’s estimation, so it may not agree with whether that note is actually played. In the equation above, α is a parameter and is set to 0.3 in the current implementation.

Then, $z_{t,k,n} = m_{t,k,n}y_{t,k,n}$ is calculated. The guitar can play only one note at each string at the same time. As a result,

$$\hat{n}_{t,k} = \underset{n}{\operatorname{argmax}} z_{t,k,n}$$

is calculated and the fret $\hat{n}_{t,k}$ of string k is considered to be played at time t . Then, a MIDI Note On message is generated for the corresponding note number. However, no fret is considered to be played on string k at time t when every element of $\{z_{t,k,n}\}_n$ is lower than a certain threshold (0.3 in the current implementation).

4. EXPERIMENTS

4.1 Experiment 1 — Use of neural network

Experimental conditions

To confirm the effect of the use of the neural network described in Section 2.2, we conducted an experiment about converting guitar performances to the MIDI format. We used a Roland GK-3 installed into a Stratocaster as a MIDI guitar with a guitar synthesizer (Roland GR-55). The first author of this paper played 79 four-measure funk rhythmic phrases taken from a guitar phrase book [11]. Of these 79 phrases, those shown in Figures 2 and 3 were used for the 2nd preliminary performance. The remaining phrases were used for test data. We attempted the following three cases:

Case 1 Using only Figure 2,

Case 2 Using only Figure 3, and

Case 3 Using both Figures 2 and 3

for the 2nd preliminary performance. We used neural networks that had three to ten nodes in the hidden layer and will present only the best result.

Experimental results

The results are listed in Table 1. The number of hidden nodes was three. For brevity, we list the accuracy for each chapter instead of each phrase in [11]. In [11], phrases are divided into 16 chapters according to their playing styles, and each chapter includes several phrases. Whereas the F-score for the baseline method was 0.516 on average, the F-score for the proposed method was 0.526 in Case 3. This difference is not very large but one must consider that the proposed method acquired the best threshold because the

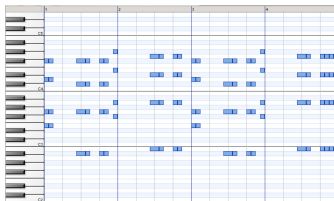


Figure 2. Phase 1 for the 2nd preliminary performance



Figure 3. Phase 2 for the 2nd preliminary performance

listed result for the baseline method was the best one in given various thresholds.

Figure 4 shows an example of the experimental results. While the baseline method generated many false positives, especially in the first measure, most of these false positives were eliminated by the proposed method. Thus, the precision rate was improved from 0.659 to 0.692. However, some positives were eliminated so the recall rate decreased slightly (from 0.562 to 0.556).

Figure 5 shows another example. Whereas the baseline method generated many false negatives from the beginning to the end, such false negatives were eliminated by the proposed method. The precision rate was improved from 0.465 to 0.661.

4.2 Experiment 2 — Integration

Experimental conditions

To confirm the effect of the integration described in Section 3, we conducted on audio-to-MIDI conversion of guitar performances using the MIDI guitar only (MGT), the NMF and neural network only (NMF+NN; Section 2.2), and their integration (INT; Section 3). We used the same data as Experiment 1. For the 2nd preliminary performance, we used both Figures 2 and 3 (Case 3). Also in this experiment, we used neural networks that had three to ten nodes in the hidden layer, and will present only the best result for each condition.

Experimental results

The results are listed in Table 2. The numbers of hidden nodes were three for NMF+NN and nine for INT. Whereas the precision rates for MGT and NMF+NN were 0.258 and 0.513, respectively, the precision rate improved to 0.660

Table 1. Result of Experiment 1 (*R*: recall rates, *P*: precision rates, *F*: F-score)

Chapters	Baeline method (Simple thresholding)			Proposed method								
	<i>R</i>	<i>P</i>	<i>F</i>	Case 1			Case 2			Case 3		
				<i>R</i>	<i>P</i>	<i>F</i>	<i>R</i>	<i>P</i>	<i>F</i>	<i>R</i>	<i>P</i>	<i>F</i>
1	0.640	0.509	0.552	0.636	0.422	0.503	0.642	0.376	0.473	0.611	0.458	0.521
2	0.647	0.489	0.555	0.624	0.483	0.538	0.661	0.490	0.563	0.635	0.574	0.595
3	0.579	0.496	0.531	0.468	0.483	0.472	0.603	0.502	0.541	0.525	0.560	0.539
4	0.536	0.510	0.519	0.554	0.509	0.529	0.576	0.479	0.521	0.562	0.524	0.541
5	0.731	0.557	0.585	0.759	0.399	0.503	0.809	0.431	0.550	0.838	0.437	0.561
6	0.538	0.410	0.465	0.531	0.459	0.491	0.581	0.427	0.491	0.539	0.460	0.496
7	0.580	0.601	0.465	0.562	0.624	0.569	0.640	0.569	0.590	0.636	0.668	0.635
8	0.528	0.577	0.540	0.499	0.481	0.488	0.544	0.520	0.528	0.518	0.536	0.525
9	0.401	0.489	0.431	0.415	0.418	0.413	0.425	0.423	0.422	0.416	0.451	0.430
10	0.464	0.403	0.429	0.476	0.346	0.397	0.472	0.326	0.376	0.442	0.350	0.382
11	0.432	0.621	0.505	0.445	0.621	0.510	0.466	0.589	0.516	0.470	0.643	0.535
12	0.313	0.600	0.411	0.343	0.502	0.407	0.373	0.429	0.399	0.399	0.567	0.468
13	0.621	0.527	0.541	0.568	0.505	0.504	0.652	0.522	0.559	0.611	0.541	0.543
14	0.412	0.442	0.422	0.395	0.425	0.407	0.456	0.425	0.436	0.441	0.473	0.451
15	0.576	0.407	0.474	0.460	0.362	0.384	0.558	0.521	0.463	0.520	0.554	0.418
Final	0.475	0.413	0.422	0.480	0.400	0.424	0.485	0.377	0.415	0.484	0.416	0.437
Average	0.530	0.503	0.516	0.513	0.465	0.488	0.559	0.463	0.506	0.540	0.513	0.526



Figure 4. Example of Experiment 1 (Track 47-2)

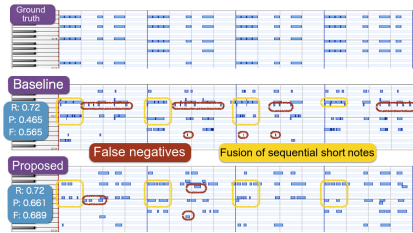


Figure 5. Example of Experiment 1 (Track 09-1)

via the integration. This fact arises because many false positives were eliminated in INT. The recall rate for INT was 0.595; that for MGT was 0.528. The recall rate improved because sequential short notes were fused in MGT, as will be illustrated below, but such errors rarely appeared in INT. Accordingly, the F-score for INT was 0.626; the F-scores for MGT and NMF+NN were 0.347 and 0.526, respectively.

Focusing on the results for each chapter, we can see that the recall rates for 11 chapters (Chapters 1, 2, 3, 4, 6, 7, 8, 10, 12, 13, and final) was improved compared with MGT. However, for other chapters (Chapters 5, 9, 11, 14, and 15) the recall rates decreased. Chapter 5 in [11] features monophonic phrases but the 2nd preliminary performance did not include monophonic phrases. This mismatch is why the recall rate decreased in Chapter 5. The phrases in Chapters 9, 14, and 15 also included monophonic notes.

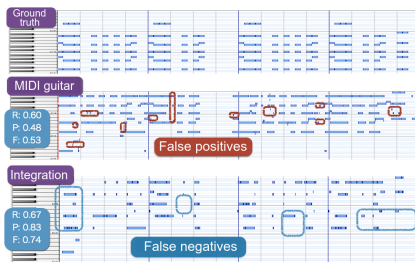
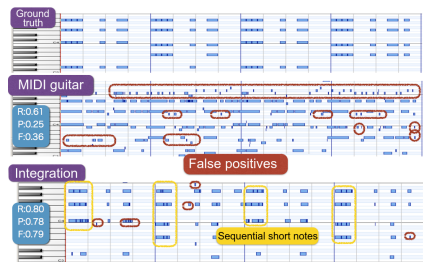
On the other hand, the precision rate improved for every chapter compared with MGT. In particular, the precision rate improved by more than 0.5 for Chapters 5, 11, and 13.

Figure 6 shows an example of the results. While MGT generated false positives in the whole phrase, such false positives were eliminated in INT as described above. Thus, the precision rate significantly improved from 0.48 (MGT) to 0.83 (INT). At the same time, however, some true positives were also eliminated. On the other hand, MGT caused errors in the fusion of sequential short notes; INT reduced such errors. Eventually, the recall rate increased from 0.60 (MGT) to 0.67 (INT).

Figure 7 shows another example. Similar to the data shown in Figure 6, MGT resulted in errors due to the fusion of sequential short notes at the first beat of every measure. In INT, such errors were corrected. Thus, the recall rate was improved from 0.61 (MGT) to 0.80 (INT). In addition, MGT generated false positives from the second half of the first measure to the last measure; these false positives were eliminated in INT. Thus, the precision rate improved from 0.25 (MGT) to 0.78 (INT). Accordingly, the F-score improved from 0.36 (MGT) to 0.79 (INT).

Table 2. Result of Experiment 2 (*R*: recall rates, *P*: precision rates, *F*: F-score)

Chapters	MIDI guitar (MGT)			NMF+NN			Integration (INT)		
	<i>R</i>	<i>P</i>	<i>F</i>	<i>R</i>	<i>P</i>	<i>F</i>	<i>R</i>	<i>P</i>	<i>F</i>
1	0.668	0.445	0.513	0.611	0.458	0.521	0.758	0.589	0.660
2	0.630	0.230	0.338	0.635	0.574	0.595	0.717	0.648	0.679
3	0.380	0.310	0.327	0.525	0.560	0.539	0.763	0.613	0.679
4	0.505	0.233	0.313	0.562	0.524	0.541	0.508	0.670	0.575
5	0.805	0.110	0.195	0.838	0.437	0.561	0.731	0.643	0.675
6	0.583	0.187	0.287	0.539	0.460	0.496	0.649	0.660	0.641
7	0.493	0.205	0.278	0.636	0.668	0.635	0.597	0.660	0.606
8	0.503	0.215	0.295	0.518	0.536	0.525	0.593	0.644	0.602
9	0.533	0.345	0.408	0.416	0.451	0.430	0.398	0.603	0.469
10	0.463	0.190	0.267	0.442	0.350	0.382	0.732	0.545	0.623
11	0.425	0.345	0.375	0.470	0.643	0.535	0.378	0.825	0.493
12	0.310	0.260	0.285	0.399	0.567	0.468	0.488	0.708	0.574
13	0.438	0.183	0.250	0.611	0.541	0.543	0.640	0.674	0.644
14	0.555	0.280	0.360	0.441	0.473	0.451	0.475	0.776	0.550
15	0.652	0.354	0.434	0.520	0.554	0.418	0.532	0.695	0.517
Final	0.505	0.238	0.313	0.484	0.416	0.437	0.555	0.607	0.542
Average	0.528	0.258	0.347	0.540	0.513	0.526	0.595	0.660	0.626

**Figure 6.** Example of Experiment 2 (Track 47-2)**Figure 7.** Example of Experiment 2 (Track 09-2)

5. CONCLUSION

A MIDI guitar is a promising tool for guitarists and therefore is being sold by electronic musical instrument manufacturers. However, the audio-to-MIDI conversion accuracy of MIDI guitars is still insufficient. In particular, the accuracy is very low for phrases including many brushing notes like those used in our experiments. To improve this accuracy, we attempted to integrate the output of the MIDI guitar and the signal processing result of the guitar's audio output. Our experimental results showed a significant improvement in accuracy: the F-score was 0.626 compared with 0.347 for the MIDI guitar only.

Although this improvement is significant, we need to improve the accuracy even more to ensure practical use of MIDI guitars. An idea for further improvement may be increasing quantity of training data for the neural network (i.e., the 2nd preliminary performance). However, increasing these data will result in an increase in the user's labor and the time required for learning the neural network.

We will therefore investigate a reasonable tradeoff between these time investments and the outcome. In addition, we will assess the latency in outputting MIDI messages because this latency is an important factor in the use of MIDI guitars as musical instruments.

Acknowledgment: This work was supported by JSPS KAKENHI Grant Number 26240025. Also, we thank the members of this JSPS KAKENHI project including Prof. Shigeiki Sagayama and Prof. Gen Hori for their fruitful suggestions.

6. REFERENCES

- [1] K. Arimoto, T. Fujishima, and M. Goto. A multiple F0 estimation method using specific harmonic structure models for guitar performances (in Japanese). In *Proc. 2006 Autumn Meeting of Acoustic Society of Japan*, pages 585–586. 2006.
- [2] A. M. Barbancho and A. Klapuri. Automatic transcription of guitar chords and fingering from audio. In *IEEE Transactions on Audio, Speech, and Language Processing*, volume 20, pages 915–921. 2012.
- [3] X. Fiss and A. Kwasinski. Automatic real-time electric gui-

- tar audio transcription. In *Proc. IEEE-ICASSP 2011*, pages 373–376, 2011.
- [4] M. Goto. A real-time music-scene-description system: Predominant-F0 estimation for detecting melody and bass lines in real-world audio signals. *Speech Comm.*, 43(4):311–329, 2004.
- [5] J. Hartquist. Real-time musical analysis of polyphonic guitar audio. Master's thesis, The Faculty of California Polytechnic State University, 2012.
- [6] A. B. Israel and T. N. E. Greville. *Generalized Inverses: Theory and Applications*. Springer, 2003.
- [7] D. D. Lee and H. S. Seung. Learning the parts of objects with nonnegative matrix factorization. *Nature*, 401:788–791, 1999.
- [8] P. D. O'Grady and S. T. Rickard. Automatic hexaphonic guitar transcription using non-negative constraints. In *Proc. IEEE-ISSC 2009*, 2009.
- [9] M. Paleari, B. Huet, A. Schutz, and D. Slock. A multimodal approach to music transcription. In *Proc. IEEE-ICIP 2008*, pages 93–96, 2008.
- [10] T. Yamagami and K. Itou. A bimodal music dictation method for composition support by using guitar performance video (in Japanese). In *Proc. of IPSJ National Convention 2014*, volume 2, pages 365–366, 2014.
- [11] K. Yamaguchi. *16 beat ga minitsuku! Funk de oboeru otona no cutting (in Japanese)*. Rittor Music, 2013.
- [12] K. Yazawa, K. Itoyama, and H. G. Okuno. Automatic transcription of guitar tablature from audio signals in accordance with player's proficiency. In *Proc. IEEE-ICASSP 2014*, pages 3146–3150, 2014.

ANALYSIS OF INTONATION TRAJECTORIES IN SOLO SINGING

Jiajie Dai, Matthias Mauch, Simon Dixon

Centre for Digital Music, Queen Mary University of London, United Kingdom
{j.dai, m.mauch, s.e.dixon}@qmul.ac.uk

ABSTRACT

We present a new dataset for singing analysis and modelling, and an exploratory analysis of pitch accuracy and pitch trajectories. Shortened versions of three pieces from *The Sound of Music* were selected: “Edelweiss”, “Do-Re-Mi” and “My Favourite Things”. 39 participants sang three repetitions of each excerpt without accompaniment, resulting in a dataset of 21762 notes in 117 recordings. To obtain pitch estimates we used the *Tony* software’s automatic transcription and manual correction tools. Pitch accuracy was measured in terms of pitch error and interval error. We show that singers’ pitch accuracy correlates significantly with self-reported singing skill and musical training. Larger intervals led to larger errors, and the tritone interval in particular led to average errors of one third of a semitone. Note duration (or inter-onset interval) had a significant effect on pitch accuracy, with greater accuracy on longer notes. To model drift in the tonal centre over time, we present a sliding window model which reveals patterns in the pitch errors of some singers. Based on the trajectory, we propose a measure for the magnitude of drift: tonal reference deviation (TRD). The data and software are freely available.¹

1. INTRODUCTION

Singing is common in all human societies [2], yet the factors that determine singing proficiency are still poorly understood. Many aspects are important to singing, including pitch, rhythm, timbre, dynamics and lyrics; here we focus entirely on the pitch dimension. Music psychologists have studied singing pitch [4, 6, 18], and engineers have developed advanced software for automatic pitch tracking [5, 11, 21], but the process of annotating and analysing the pitch of singing data remains a laborious task. In this paper, we present a new extensive dataset for the analysis of unaccompanied solo singing, complete with audio, pitch tracks, and hand-annotated note tracks matched to the scores of the music. In addition, we provide an analysis of the data with a focus on intonation: pitch errors,

interval errors, pitch drift, and the factors that influence these phenomena.

Intonation, defined as “accuracy of pitch in playing or singing” [23], or “the act of singing or playing in tune” [12], is one of the main priorities in choir rehearsals [9] and in choral practice manuals (e.g. [3]). Good intonation involves the adjustment of pitch to maximise the consonance of simultaneous notes, but it also has a temporal aspect, particularly in the absence of instrumental accompaniment, where the initial tonal reference can be forgotten over time [15]. A cappella ensembles frequently observe a change in tuning over the duration of a piece, even when they are unable to detect any local changes. This phenomenon, called *intonation drift* or pitch drift [22], usually exhibits as a lowering of pitch, or downward drift [1]. Several studies present evidence that drift is induced by harmonic progressions as singers negotiate the tradeoff between staying in tune and singing in just intonation [7, 10, 24]. Yet this is not the only cause of drift, since drift is also observed in solo singing, such as unaccompanied solo folk songs [17] and even queries to query-by-humming systems [20]. A factor that has received relatively little attention in the singing research community is the effect of note duration on singing accuracy [8], so one of our aims in this paper is to explore the effect of duration.

The definitions of intonation given above imply the existence of a reference pitch, which could be provided by accompanying instruments or (as in the present case) could exist solely in the singer’s memory. This latter case allows for the reference to change over time, and thus explain the phenomenon of drift. We introduce a novel method to model this internal reference as the pitch which minimises the intonation error given some weighted local context, and we compare various context windows for parametrising our model. Using this model of reference pitch, we compute pitch error as the signed pitch difference relative to the reference pitch and score, measured in semitones on an equal-tempered scale. Interval error is measured on the same scale, without need of any reference pitch, and pitch drift is given by the trajectory of score-normalised reference pitch over time.

In this paper we explore which factors may explain intonation error in our singing data. The effects of four singer factors, obtained by self-report, were tested for significance. Most of the participants in this study were amateur singers without professional training. Their musical background, years of training, frequency of practice and self-reported skill were all found to have a significant effect on

¹ see Data Availability, Section 7



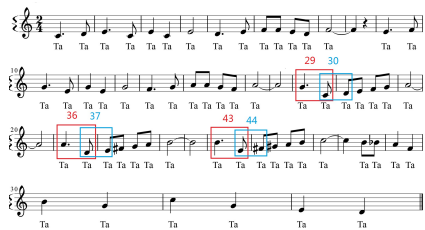


Figure 1: Score of piece Do-Re-Mi, with some intervals marked (see Section 3)

Table 1: Summary details of the three songs used in this study.

Title	Tempo (BPM)	Key	Notes
Edelweiss	80	B \flat	54
Do-Re-Mi	120	C	59
My Favourite Things	132	Em	73

intonation errors. We then considered as piece factors three melodic features, note duration, interval size and the presence of a tritone interval, for their effect on intonation. All of these features had a significant effect on both pitch and interval error. Finally we consider the pitch drift trajectories of individual singers. Our model tracks the direction and magnitude of cumulative pitch errors and captures how well participants remain in the same key. Some trajectories have periodic structure, revealing systematic errors in the singing.

2. MATERIALS AND METHODS

2.1 Musical material

We chose three songs from the musical “The Sound of Music” as our material: “Edelweiss”, “Do-Re-Mi” (shown in Figure 1) and “My Favourite Things.” Despite originating from one work, the pieces were selected as being diverse in terms of tonal material and tempo (Table 1), well-known to many singers, and yet sufficiently challenging for amateur singers. The pieces were shortened so as to contain a single verse without repeats, which the participants were asked to sing to the syllable “ta”. In order to observe long-term pitch trends, each song was sung three times consecutively. Each trial lasted a little more than 5 minutes.

2.2 Participants

We recruited 39 participants (12 male, 27 female), most of whom are members of our university’s music society or our music-technology focused research group. Some participants took part in the experiments remotely. The age of the participants ranged from 20 to 27 years (mean 23.3, median 23 years). We asked all participants to self-assess their musical background with questions loosely based on

the Goldsmiths Musical Sophistication Index [16].² Table 2 shows the results, suggesting a range of skill levels, with a strong bias towards amateur singers.

Table 2: Self-reported musical experience

Musical Background	Instrumental Training		
None	5	None	5
Amateur	27	1–2 years	15
Semi-professional	5	3–4 years	7
Professional	2	5+ years	12
Singing Skill		Singing Practice	
Poor	2	None	4
Low	25	Occasionally	22
Medium	9	Often	12
High	3	Frequently	1

2.3 Recording procedure

Participants were asked to sing each piece three times on the syllable ‘ta’. They were given the starting note but no subsequent accompaniment, except unpitched metronome clicks.

2.4 Annotation

We used the software *Tony*³ to annotate the notes in the audio files [13]: pitch track and notes were extracted using the pYIN algorithm [14] and then manually checked and, if necessary, corrected. Approximately 28 corrections per recording were necessary; detailed correction metrics on this data have been reported elsewhere [13].

2.5 Pitch metrics

The *Tony* software outputs the median fundamental frequency f_0 for every note. We relate fundamental frequency to musical pitch p as follows:

$$p = 69 + 12 \log_2 \frac{f_0}{440 \text{ Hz}} \quad (1)$$

This scale is chosen such that a difference of 1 corresponds to 1 semitone. For integer values of p the scale coincides with MIDI pitch numbers, with reference pitch A4 tuned to 440 Hz ($p = 69$).

2.5.1 Interval Error

A musical interval is the difference between two pitches [19] (which is proportional to the logarithm of the ratio of the fundamental frequencies of the two pitches). Using Equation 1, we define the interval from a pitch p_1 to the pitch p_2 as $i = p_2 - p_1$ and hence we can define the interval error between a sung interval i and the expected nominal interval i_n (given by the musical score) as:

$$e^{int} = i - i_n \quad (2)$$

² The questions were: How do you describe your musical background? How many years do you have instrument training? How do you describe your singing skills? How often do you practice your singing skills?

³ <https://code.soundsoftware.ac.uk/projects/tony>

Hence, for a piece of music with M intervals $\{e_1^{int}, \dots, e_M^{int}\}$, the mean absolute interval error (MAIE) is calculated as follows:

$$\text{MAIE} = \frac{1}{M} \sum_{i=1}^M |e_i^{int}| \quad (3)$$

2.5.2 Tonal reference curves and pitch error

In unaccompanied singing, pitch error is ill-defined, since singers use intonation with respect to their internal reference, which we cannot track directly. If it is assumed that this internal reference doesn't change, we can estimate it via the mean error with respect to a nominal (or given) reference pitch. However, it is well-known that unaccompanied singers (and choirs) do not maintain a fixed internal reference (see Section 1). Previously, this has been addressed by estimating the singer's reference frequency using linear regression [15], but as there is no good reason to assume that drift is linear, we adopt a sliding window approach in order to provide a local estimate of tuning reference.

The first step is to take the annotated musical pitches p_i of a recording and remove the nominal pitch s_i given by the score, $t_i^* = p_i - s_i$, which we adjust further by subtracting the mean: $t_i = t_i^* - \bar{t}^*$. The resulting raw tonal reference estimates t_i are then used as a basis for our tonal reference curves and pitch error calculations.

The second step is to find a smooth trajectory based on these raw tonal reference estimates. For each note, we calculate the weighted mean of t_i in a context window around the note, obtaining the reference pitch c_i , from which the pitch error can be calculated:

$$c_i = \sum_{k=-n}^n w_k t_{i+k}, \quad (4)$$

where $\sum_{k=-n}^n w_k = 1$. Any window function $W = \{w_k\}$ can be used in Equation 4. We experimented with symmetric windows with two different window shapes (rectangular and triangular) and seven window sizes (3, 5, 7, 9, 11, 15 and 25 notes) to arrive at smooth tonal reference curves. The rectangular window $W^{R,N} = \{w_k^{R,N}\}$ centred at the i^{th} note is used to calculate the mean of its N -note neighbourhood, giving the same weight to all notes in the neighbourhood, but excluding the i^{th} note itself:

$$w_k^{R,N} = \begin{cases} \frac{1}{N-1}, & 1 \leq |k| \leq \frac{N-1}{2} \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

The triangular window $W^{T,N} = \{w_k^{T,N}\}$ gives more weight to notes near the i^{th} note (while still excluding the i^{th} note itself). For example, if the window size is 5, then the weights are proportional to 1, 2, 0, 2, 1. More generally:

$$w_k^{T,N} = \begin{cases} \frac{2N+2-4|k|}{N^2-1}, & 1 \leq |k| \leq \frac{N-1}{2} \\ 0, & \text{otherwise.} \end{cases} \quad (6)$$

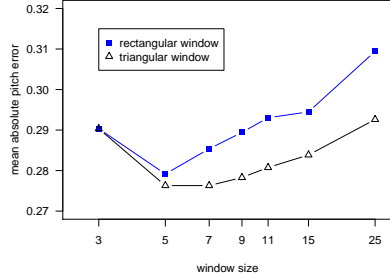


Figure 2: Pitch error (MAPE) for different sliding windows.

The smoothed tonal reference curve c_i is the basis for calculating the pitch error:

$$e_i^p = t_i - c_i, \quad (7)$$

so for a piece with M notes with associated pitch errors e_1^p, \dots, e_M^p , the mean absolute pitch error (MAPE) is:

$$\text{MAPE} = \frac{1}{M} \sum_{i=1}^M |e_i^p|. \quad (8)$$

2.5.3 Tonal reference deviation

The tonal reference curves c_i can also be used to calculate a new measure of the extent of fluctuation of a singer's reference pitch. We call this measure tonal reference deviation (TRD), calculated as the standard deviation:

$$\text{TRD} = \sqrt{\frac{1}{M-1} \sum_{i=1}^M (c_i - \bar{c}_M)^2}. \quad (9)$$

3. RESULTS

We first compare multiple choices of window for the calculation of the smoothed tonal reference curves c_i (Section 2.5.2), which provide the local tonal reference estimate used for calculating mean absolute pitch error (MAPE). We assume that the window that gives rise to the lowest MAPE models the data best. Figure 2 shows that for both window shapes an intermediate window size N of 5 notes minimises MAPE, with the triangular window working best (MAPE = 0.276 semitones, computed over all singers and pieces). Hence, we use this window for all further investigations relating to pitch error, including tonal reference curves, and for understanding how pitch error is linked to note duration and singers' self-reported skill and experience.

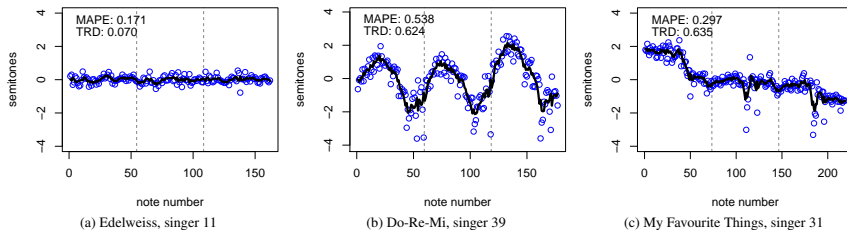


Figure 3: Examples of tonal reference trajectories. Dashed vertical lines delineate the three repetitions of the piece.

3.1 Smoothed tonal reference curves

The smoothed curves exhibit some unexpected behaviour. Figure 3 shows three examples of different participants and pieces. Several patterns emerge. Figure 3a shows a performance in which pitch error is kept within half a semitone and tonal reference is almost completely stable. This is reflected in very low values of MAPE (0.171) and TRD (0.070), respectively. However, most singers' tonal reference curves fluctuate. For example, Figure 3b illustrates a tendency of some singers to smoothly vary their pitch reference in direct response to the piece. The trajectory shows a periodic structure synchronised with the three repetitions of the piece. The fluctuation measure TRD is much higher as a result (0.624). This is a common pattern we have observed. The third example (Figure 3c) illustrates that strong fluctuations are not necessarily periodic. Here, TRD (0.635) is nearly identical, but originates from a mostly consistent downward trajectory. The singer makes significant errors in the middle of each run of the piece, most likely due to the difficult interval of a downward tritone occurring twice (notes 42 and 50; more discussion below). Comparing Figures 3b and 3c also shows that MAPE and TRD are not necessarily related. Despite large fluctuations (TRD) in both, pitch error (MAPE) is much smaller in Figure 3c (0.297).

Turning from the trajectories to pitch error measurements, we observe that the three pieces show distinct patterns (Figure 4). The first piece, Edelweiss, appears to be the easiest to sing, with relatively low median pitch errors. In Do-Re-Mi, the third quarter of the piece appears much more difficult than the rest. This is most likely due to faster runs and the presence of accidentals, taking the singer out of the home tonality. Finally, My Favourite Things exhibits a very distinct pattern, with relatively low pitch errors throughout, except for one particular note (number 50), which is reached via a downward tritone, a difficult interval to sing. The same tritone (A-D \sharp) occurs at note 42, where the error is smaller and notably in the opposite direction (this D \sharp is flat, while note 50 is over a semitone sharp on average). It appears that singers are drawn towards the more consonant (and more common) perfect fifth and fourth intervals, respectively.

	Estimate	Std. Err.	<i>t</i>	<i>p</i>
(intercept)	0.374	0.012	32.123	0.000
nominal duration	-0.073	0.004	-17.487	0.000
prev. nom. IOI	-0.021	0.004	-4.646	0.000
abs(nom. interval)	0.016	0.001	13.213	0.000
abs(next nom. interval)	0.010	0.001	8.471	0.000
tritone	0.370	0.019	19.056	0.000
quest. score	-0.011	0.001	-9.941	0.000

(a) MAPE

	Estimate	Std. Err.	<i>t</i>	<i>p</i>
(intercept)	0.481	0.015	33.124	0.000
nominal duration	-0.076	0.005	-14.570	0.000
prev. nom. IOI	-0.050	0.006	-8.984	0.000
abs(nom. interv.)	0.030	0.002	19.700	0.000
abs(next nom. interv.)	-0.006	0.002	-3.826	0.000
tritone	0.373	0.024	15.404	0.000
quest. score	-0.012	0.001	-8.665	0.000

(b) MAIE

Table 3: Effects of multiple covariates on error for a linear model. *t* denotes the test statistic. The *p* value rounds to zero in all cases, indicating statistical significance.

3.2 Duration, interval and proficiency factors

The observations on pitch error patterns suggest that note duration and the tritone interval may have significant impact on pitch error. In order to investigate their impact we make use of a linear model, taking into account furthermore the size of the intervals sung and singer bias via considering the singers' self assessment.

Table 3a lists all dependent variables, estimates of their effects and indicators of significance. In the following we will simply speak of how these variables influence, reduce or add to error, noting that our model gives no indication of true causation, only of correlation. We turn first to the question of whether note duration influences pitch error. The intuition is that longer notes, and notes with a longer preparation time (previous inter-onset interval, IOI), should be sung more correctly. This is indeed the case. We observe a reduction of pitch error of 0.073 semitones per added second of duration. The IOI between previous and current note also reduces pitch error, but by a smaller factor (0.021 semitones per second). Conversely, absolute nominal interval size adds to absolute pitch error, by about 0.016 semitones per interval-semitone, as does

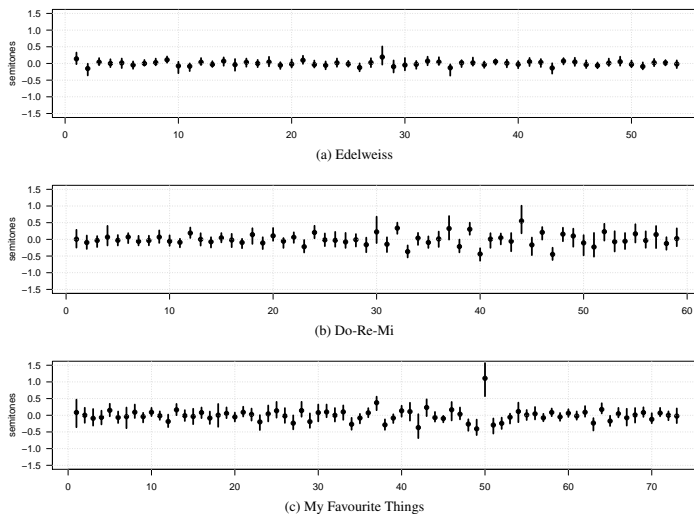


Figure 4: Pitch errors by note for each of the three pieces. The plots show the median values with bars extending to the first and third quartiles.

the absolute size of the next interval (0.010 semitones). The intuition about the tritone interval is confirmed here, as the presence of any tritone (whether upward or downward) adds 0.370 semitones—on average—to the absolute pitch error. The last covariate, questionnaire score, is the sum of the points obtained from the four self-assessment questions, with values ranging between 5 and 14. The result shows that there is correlation between the singers’ self-assessment and their absolute pitch error. For every additional point in the score their absolute pitch error is reduced by 0.012 semitones. The picture is very similar as we do the same analysis for absolute interval error (Table 3b): the effect directions of the variables are the same.

4. DISCUSSION

We have investigated how note length relates to singing accuracy, finding that notes are sung more accurately as the singer has more time to prepare and sing them. Yet it is not entirely clear what this improvement is based upon. Do longer notes genuinely give singers more time to find the pitch, or is part of the effect we observe due to measurement or statistical artefacts? To find out, we will need to examine pitch at the sub-note level, taking vibrato and note transitions into account. Conversely, studying the effect of melodic context on the underlying pitch track could shed light on the physical process of singing, and could be used for improved physical modelling of singing.

Overall, the absolute pitch error of singers (mean: 28 cents; median: 18; std.dev.: 36) and the absolute inter-

val error (mean: 34 cents; median: 22; std.dev.: 46) are slightly higher than those reported elsewhere [15], but this may reflect the greater difficulty of our musical material in comparison to “Happy Birthday”. We also did not exclude singers for their pitch errors, although the least accurate singers had MAPE and MAIE values of more than half a semitone, i.e. they were on average closer to an erroneous note than to the correct one. That the values of MAIE and MAPE are similar is to be expected, as interval error is the limiting case of pitch error, using a minimal window containing only the current and previous note.

We used a symmetric window in this work, but this could easily be replaced with a causal (one-sided) window [15], which would also be more plausible psychologically, as the singer’s internal pitch reference in our model is based equally on past sung notes and future not-yet-sung notes. However, for post hoc analysis, the fuller context might reveal more about the singer’s internal state (which must influence the future tones) than the more restricted causal model.

Figure 4 shows how the three pieces in our data differ in terms of pitch accuracy. It is interesting to see that accidentals (which result in a departure from the established key), and the tritone as a particular example, seem to have a strong adverse impact on accuracy. To compile more detailed statistical analyses like the ones in Table 3 one could conduct singing experiments on a wider range of intervals, isolated from the musical context of a song. In future work we also intend to explore the interaction between singers as they negotiate a common tonal reference.

Finally, we would like to mention that some singers took prolonged breaks between runs in a three-run rendition of a song. The recording was stopped, but no new reference note was played, so the singers resumed with the memory of what they last sung. As part of the reproducible code package (see Section 7) we provide information on which recordings were interrupted and at which break. We found that the regression coefficients (Tables 3b and 3a) did not substantially change as a result of these interruptions.

5. CONCLUSIONS

We have presented a new dataset for singing analysis, investigating the effects of singer and piece factors on the intonation of unaccompanied solo singers. Pitch accuracy was measured in terms of pitch error and interval error. We introduced a new model of tonal reference computed using the local neighbourhood of a note, and found that a window of two notes each side of the centre note provides the best fit to the data in terms of minimising the pitch error. The temporal evolution of tonal reference during a piece revealed patterns of tonal drift in some singers, others appeared random, yet others showed periodic structure linked to the score. As a complement to errors of individual notes or intervals, we introduced a measure for the magnitude of drift, tonal reference deviation (TRD), and illustrated how it behaves using several examples.

Two types of factors influencing pitch error were investigated, those related to the singers and those related to the material being sung. In terms of singer factors, we found that pitch accuracy correlates with self-reported singing skill level, musical training, and frequency of practice. Larger intervals in the score led to larger errors, but only accounted for 2–3 cents per semitone of the mean absolute errors. On the other hand, the tritone interval accounted for 35 cents of error when it occurred, and in one case led to a large systematic error across many of the singers. We hypothesised that note duration might also have an effect on pitch accuracy, as singers make use of aural feedback to regulate their pitch, which results in less stable pitch at the beginnings of notes. This was indeed the case: a small but significant effect of duration was found for both the current note, and the nominal time taken from the onset of the previous note; longer durations led to greater accuracy. Many aspects of the data remain to be explored, such as the potential effects of scale degree, consonance, modulation, and rhythm.

6. ACKNOWLEDGEMENTS

Matthias Mauch is funded by a Royal Academy of Engineering Research Fellowship. Many thanks to all the participants who contributed their help during this project.

7. DATA AVAILABILITY

All audio recordings analysed here (and corresponding trajectory plots) can be obtained from <http://dx.doi.org/10.6084/m9.figshare.1482221>.

The code and the data needed to reproduce our results (note annotations, questionnaire results, interruption details) are provided in an open repository at <https://code.soundsoftware.ac.uk/projects/dai2015analysis-resources>.

8. REFERENCES

- [1] P. Alldahl. *Choral Intonation*. Gehrman, Stockholm, Sweden, 2006. p. 4.
- [2] D.E. Brown. *Human Universals*. Temple University Press, Philadelphia, 1991.
- [3] D. S. Crowther. *Key Choral Concepts: Teaching Techniques and Tools to Help Your Choir Sound Great*. Cedar Fort, 2003.
- [4] S. Dalla Bella, J. Giguère, and I. Peretz. Singing proficiency in the general population. *Journal of the Acoustical Society of America*, 121(2):1182, 2007.
- [5] A. de Cheveigné and H. Kawahara. YIN, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.
- [6] J. Devaney and D. P. W. Ellis. An empirical approach to studying intonation tendencies in polyphonic vocal performances. *Journal of Interdisciplinary Music Studies*, 2(1&2):141–156, 2008.
- [7] J. Devaney, M. Mandel, and I. Fujinaga. A study of intonation in three-part singing using the automatic music performance analysis and comparison toolkit (AM-PACT). In *13th International Society of Music Information Retrieval Conference*, pages 511–516, 2012.
- [8] J. Fyk. Vocal pitch-matching ability in children as a function of sound duration. *Bulletin of the Council for Research in Music Education*, pages 76–89, 1985.
- [9] C. M. Ganschow. Secondary school choral conductors’ self-reported beliefs and behaviors related to fundamental choral elements and rehearsal approaches. *Journal of Music Teacher Education*, 20(10):1–10, 2013.
- [10] D. M. Howard. Intonation drift in a capella soprano, alto, tenor, bass quartet singing with key modulation. *Journal of Voice*, 21(3):300–315, May 2007.
- [11] H. Kawahara, J. Estill, and O. Fujimura. Aperiodicity extraction and control using mixed mode excitation and group delay manipulation for a high quality speech analysis, modification and synthesis system STRAIGHT. In *Proceedings of the Workshop on Models and Analysis of Vocal Emissions for Biomedical Applications (MAVEBA)*, pages 59–64, 2001.
- [12] M. Kennedy. *The Concise Oxford Dictionary of Music*. Oxford University Press, Oxford, United Kingdom, 1980. p. 319.

- [13] M. Mauch, C. Cannam, R. Bittner, G. Fazekas, J. Salamon, J. Bello, J. Dai, and S. Dixon. Computer-aided melody note transcription using the Tony software: Accuracy and efficiency. In *Proceedings of the First International Conference on Technologies for Music Notation and Representation (TENOR 2015)*, pages 23–30, 2015.
- [14] M. Mauch and S. Dixon. pYIN: A fundamental frequency estimator using probabilistic threshold distributions. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, pages 659–663, 2014.
- [15] M. Mauch, K. Frieler, and S. Dixon. Intonation in unaccompanied singing: Accuracy, drift, and a model of reference pitch memory. *Journal of the Acoustical Society of America*, 136(1):401–411, 2014.
- [16] D. Müllensiefen, B. Gingras, and L. Stewart. Piloting a new measure of musicality: The Goldsmiths’ Musical Sophistication Index. Technical report, Goldsmiths, University of London, 2011.
- [17] M. Müller, P. Grosche, and F. Wiering. Automated analysis of performance variations in folk song recordings. In *Proceedings of the International Conference on Multimedia Information Retrieval*, pages 247–256, 2010.
- [18] P. Q. Pfordresher and S. Brown. Poor-pitch singing in the absence of “tone deafness”. *Music Perception*, 25(2):95–115, 2007.
- [19] E. Prout. *Harmony: Its Theory and Practice*. Cambridge University Press, 2011.
- [20] M. P. Ryyänen. Probabilistic modelling of note events in the transcription of monophonic melodies. Master’s thesis, Tampere University of Technology, Finland, 2004. pp. 27–30.
- [21] J. Salamon, E. Gómez, D. P. W. Ellis, and G. Richard. Melody extraction from polyphonic music signals: Approaches, applications, and challenges. *IEEE Signal Processing Magazine*, 31(2):118–134, 2014.
- [22] R. Seaton, D. Pim, and D. Sharp. Pitch drift in A Cappella choral singing. *Proceedings of the Institute for Acoustics Annual Spring Conference*, 35(1):358–364, 2013.
- [23] J. Swannell. *The Oxford Modern English Dictionary*. Oxford University Press, USA, 1992. p. 560.
- [24] H. Terasawa. Pitch Drift in Choral Music, 2004. Music 221A final paper, URL <https://ccrma.stanford.edu/~hiroko/pitchdrift/paper221A.pdf>.

EVALUATING THE *GENERAL CHORD TYPE* REPRESENTATION IN TONAL MUSIC AND ORGANISING GCT CHORD LABELS IN FUNCTIONAL CHORD CATEGORIES

Maximos Kaliakatsos-Papakostas, Asterios Zacharakis, Costas Tsougras, Emiliios Cambouroopoulos

School of Music Studies, Aristotle University of Thessaloniki, Greece

{emiliios,maxk,tsougras,aszachar}@mus.auth.gr

ABSTRACT

The General Chord Type (GCT) representation is appropriate for encoding tone simultaneities in any harmonic context (such as tonal, modal, jazz, octatonic, atonal). The GCT allows the re-arrangement of the notes of a harmonic sonority such that abstract idiom-specific types of chords may be derived. This encoding is inspired by the standard roman numeral chord type labelling and is, therefore, ideal for hierarchic harmonic systems such as the tonal system and its many variations; at the same time, it adjusts to any other harmonic system such as post-tonal, atonal music, or traditional polyphonic systems. In this paper the descriptive potential of the GCT is assessed in the tonal idiom by comparing GCT harmonic labels with human expert annotations (Kostka & Payne harmonic dataset). Additionally, novel methods for grouping and clustering chords, according to their GCT encoding and their functional role in chord sequences, are introduced. The results of both harmonic labelling and functional clustering indicate that the GCT representation constitutes a suitable scheme for representing effectively harmony in computational systems.

1. INTRODUCTION

Computational systems developed for harmonic analysis and/or harmonic generation (e.g. melodic harmonisation), rely on chord labelling schemes that are relevant and characteristic of particular idioms [7, 10, 20, 21, 26]. There exist various typologies for encoding note simultaneities that embody different levels of harmonic information/abstraction and cover different harmonic idioms. For instance, some commonly used chord notations in tonal music are the following: figured bass (pitch classes denoted above a bass note – no concept of ‘chord’), popular music guitar style notation or jazz notation (absolute chord), roman numeral encoding (relative chord to a key) [18] - see, Harte’s [12]

formal tonal chord symbol representation. For atonal and other non-tonal systems, pitch-class set theoretic encodings [8] may be employed. There exists no single chord encoding scheme that can be applied to all harmonic systems with sufficient expressiveness.

Preliminary studies on the General Chord Type (GCT) [3] representation (e.g. for probabilistic melodic harmonisation [15]) indicate that it can be used both as a means to represent accurately harmonic chords and to describe musically meaningful relations between different harmonic labels in diverse music idioms. The GCT provides accurate harmonic representation in a sense that it encompasses all the pitch-class-related information about chords. At the same time, for every pitch class simultaneously the GCT algorithm rearranges pitch classes so that it identifies a root pitch class and a chord base type and extension, leading to chord representations that convey musical meaning for diverse music idioms.

It is true that the main strength of the GCT representation is its application in non-tonal harmonic idioms; some such preliminary examples have been presented in [2, 3, 14]. This paper, however, focuses on the tonal idiom, as this provides a well-studied system with reliable ground truth data against which a chord labelling and grouping algorithm can be tested. If the GCT representation can cope with such a sophisticated hierarchical harmonic system as the tonal system, then it seems likely that it can deal with other non-tonal systems (even though other simpler representations may also be adequate). Applying and testing the GCT on other musics is part of ongoing research.

The paper at hand addresses two issues regarding the GCT representation. First, an evaluation of the GCT’s ability to label chords is performed by comparing the chord roots and types it produces with human expert annotations (roman-numeral analysis) on the Kostka & Payne dataset. This analysis provides clear indications about the interpretational efficiency of the GCT (around 92% agreement with human annotations). Secondly, a grouping process is proposed, which allows the identification of the functional role of chord groups in GCT form. An initial grouping stage, solely based on the GCT expression of the chords, allows in a second stage, the identification of functional similarities according to first-order transitions of GCT chord groups. The results of this analysis on a set of Bach Chorales indicate that the functional role of GCT chord groups is determined in a reliable manner, agreeing with theoretic



© Maximos Kaliakatsos-Papakostas, Asterios Zacharakis, Costas Tsougras, Emiliios Cambouroopoulos.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Maximos Kaliakatsos-Papakostas, Asterios Zacharakis, Costas Tsougras, Emiliios Cambouroopoulos. “EVALUATING THE *GENERAL CHORD TYPE* REPRESENTATION IN TONAL MUSIC AND ORGANISING GCT CHORD LABELS IN FUNCTIONAL CHORD CATEGORIES”, 16th International Society for Music Information Retrieval Conference, 2015.

functional characteristics of chords in this idiom.

2. THE GENERAL CHORD TYPE REPRESENTATION

Harmonic analysis is a rather complex musical task that involves not only finding roots and labelling chords within a key, but also segmentation (points of chord change), identification of non-chord notes, metric information and more generally musical context [27]. In this section, we focus on the core problem of labelling chords within a given pitch hierarchy (e.g. key). We assume, for simplicity, that a full harmonic reduction (main harmonic notes) is available as input to the model along with key/modulation annotations. It is suggested that the GCT representation scheme can be used in the future so as to facilitate the harmonic reduction per se of an unreduced musical surface (e.g. by identifying dissonant chord extensions in relation to a chord's consonant base).

The General Chord Type (GCT) representation, allows the re-arrangement of the notes of a harmonic simultaneity such that a maximal consonant part determines the base of the chord, and the rest of the dissonant notes form the chord extension; the lowest note of the base is the root of the chord. The GCT representation has common characteristics with the stack-of-thirds and the virtual pitch root finding methods for tonal music, but has differences as well (see [3]). This encoding is inspired by the standard roman numeral chord type labelling, but is more general and flexible. A brief description of merely the GCT core algorithm is presented below (due to space limitations); a more extended discussion on the background concepts necessary for the GCT model as well as a more detailed description of the GCT representation are presented in [3].

2.1 Description of the GCT Algorithm

Given a classification of intervals into consonant/dissonant (binary values) and an appropriate scale background (i.e. scale with tonic), the GCT algorithm computes, for a given multi-tone simultaneity, the 'optimal' ordering of pitches such that a maximal subset of consonant intervals appears at the 'base' of the ordering (left-hand side) in the most compact form; the rest of the notes that create dissonant intervals to one or more notes of the chord 'base' form the chord 'extension'. Since a tonal centre (key) is given, the position within the given scale is automatically calculated.

Input to the algorithm is the following:

- Consonance vector: a Boolean 12-dimensional vector is employed indicating the consonance of pitch-class intervals (from 0 to 11). E.g., the vector [1, 0, 0, 1, 1, 0, 1, 1, 1, 0, 0] means that the unison, minor and major third, perfect fourth and fifth, minor and major sixth intervals are consonant – dissonant intervals are the seconds, sevenths and the tritone; this specific vector is referred to in this text as the tonal consonance vector.
- Pitch Scale Hierarchy: is given in the form of scale tones and a tonic. E.g., a *D* major scale is given as:

Table 1. GCT chord labelling example

Input: <i>Bb</i> major scale: [10, [0, 2, 4, 5, 7, 9, 11]]
Input: Consonance vector: [1, 0, 0, 1, 1, 1, 0, 1, 1, 1, 0, 0]
– input [53, 63, 69, 72, 75]
– input converted to pc-set: [0, 3, 5, 9]
– maximal consonant subset: [0, 5, 9]
– rewrite in narrowest range: [5, 9, 0]
– Dissonant tone 3 goes to the end: [5, 9, 0, 3]
– Lowest tone is root, i.e. 5 (note <i>F</i>)
– Chord with root 0: [0, 4, 7, 10] (i.e., dominant seventh)
– Absolute chord: [5, [0, 4, 7, 10]] (i.e., <i>F7</i>)
– Relative position: root is 7 semitones above the tonic
<i>Bb</i>
– Chord in relative position: [7, [0, 4, 7, 10]]
– No other maximal subset exists.
Output: [7, [0, 4, 7, 10]] (i.e. <i>V7</i>)

2, [0, 2, 4, 5, 7, 9, 11], or an *A* minor pentatonic scale as: 9, [0, 3, 5, 7, 10]

- Input chord: list of pitch classes (MIDI pitch numbers modulo 12).

Algorithm 1 GCT algorithm (core) – computational pseudocode

Require: (i) the pitch scale (tonality), (ii) a vector of the intervals considered consonant, (iii) the pitch class set (pc-set) of a note simultaneity

Ensure: The roots and types of the possible chords describing the simultaneity

- 1: find all maximal subsets of pairwise consonant tones
 - 2: select maximal subsets of maximum length
 - 3: **for** all selected maximal subsets **do**
 - 4: order the pitch classes of each maximal subset in the most compact form (chord 'base')
 - 5: add the remaining pitch classes (chord 'extensions') above the highest of the chosen maximal subset's (if necessary, add octave – pitches may exceed the octave range)
 - 6: the lowest tone of the chord is the 'root'
 - 7: transpose the tones of the chord so that the lowest becomes 0
 - 8: find position of the 'root' in regards to the given tonal centre (pitch scale)
 - 9: **end for**
-

Since the aim of this algorithm is not to perform sophisticated harmonic analysis, but rather to find a practical and efficient encoding for tone simultaneities (to be used, for instance, in statistical learning and automatic harmonic generation in the context of the project COINVENT [25]), we decided to extend the algorithm so as to reach a single chord type for each simultaneity (no ambiguity) in every case. These additional steps are described in [3] and take into account overlapping of maximal subsets and avoidance of non-scale notes in the base of chord types.

An example taken from Beethoven's *Andante Favori* (Figure 1) illustrates the application of the GCT algorithm for different consonance vectors. For the tonal vector, GCT encodes classical harmony in a straightforward manner.

All instances of the tonic chord are tagged as $[0, [0, 4, 7]]$; the dominant seventh (inverted or not) is $[7, [0, 4, 7, 10]]$; the third to last chord is a minor seventh on the second degree encoded as $[2, [0, 3, 7, 10]]$; the second and fourth chord is a Neapolitan sixth chord encoded as $[1, [0, 4, 7]]$ (which means major chord on lowered second degree) with a secondary dominant in between (the pedal G flat note in the third chord is not taken into account). This way we have an encoding that is analogous to the standard roman numeral encoding (Figure 1, ‘tonal’). If the tonal context is changed to a chromatic scale context and all intervals are considered equally consonant, i.e. all entries in consonance vector are 1s, we get the second ‘atonal’ GCT analysis (Figure 1, ‘atonal’) which amounts to normal orders (not prime forms) in standard pc-set analysis. In pitch class set theory normal orders do not have roots – however, they have transposition values (T0-T11) in relation to a reference pc (normally pc 0); the normal orders with transposition values of pc-set theory are equivalent to the GCT for the atonal consonance vector. Obviously, for tonal music, this pc-set-like analysis is weak as it misses out or obscures important tonal hierarchical relationships; however, it can encode efficiently non-tonal musics. More examples from non-tonal music in [2, 3, 14].

2.2 Qualitative evaluation of the GCT in tonal music

We tested the GCT algorithm on the Kostka-Payne dataset created by David Temperley. This dataset consists of the 46 excerpts that are longer than 8 measures from the workbook accompanying Kostka and Payne’s theory textbook *Tonal Harmony*, 3rd edition (McGraw-Hill, 1995)¹. Given the local tonality (key), the GCT algorithm was applied to all the Kostka-Payne excerpts. Then, the resulting GCTs were compared to the Kostka-Payne ground truth (i.e. the roman numeral analysis included in the Instructor’s Manual, not taking into account chord inversions). From the 919 chords of the dataset, GCT successfully encodes 847 chords, and 72 chords are labelled differently. This means that the algorithm labels 92.16% of all chords correctly.

The identified mistakes can be categorised as follows:

a) Twenty three (23) mislabeled chords were diminished seventh chords $[0, 3, 6, 9]$. As explained earlier, these symmetric chords can have as their root any of the four constituent notes. In most cases these were vii^{07} chords in various inversions, referring either to the main key or to other keys as applied chords, but in some cases they were embellishing (non functional) chords.

b) Twenty two (22) half-diminished chords $[0, 3, 6, 10]$ were labelled as minor chords with added sixth $[0, 3, 7, 9]$; e.g. $[B, D, F, A]$ was re-ordered as $[D, F, A, B]$. As a consequence, all $\text{ii}^{\text{add}6/5}$ chords in minor keys were identified as $\text{iv}^{\text{add}6}$ chords, and all vii^{07} -type chords in major keys were identified as $\text{ii}^{\text{add}6}$ chords.

c) Seventeen (17) cases had a salient note missing (e.g. diminished chord without root, dominant seventh without third, half-diminished seventh without third, etc) and this

resulted in finding a wrong root; e.g. $[G\sharp, D, F]$, vii^{07} in A minor without 3rd, was identified as $[D, F, A\flat]$, i.e. as $\text{iv}^{5\flat}$; $[B, F, A]$, vii^{07} in C major without 3rd, appears as $[F, A, B]$, i.e. $\text{IV}^{5\flat}$; $[C, E, B\flat, D\flat]$, $\text{V}^{7/9}$ in F minor, is identified as $[B\flat, D\flat, F\flat, C]$, i.e. as $\text{iv}^{5\flat/9}$; $[E\flat, G, D\flat, C]$, i.e. $\text{V}^{7/13}$ in Ab major erroneously appeared as $[C, E\flat, G, D\flat]$, i.e. $\text{iii}^{0\flat}$, while $[C, E, B\flat, A\flat]$, i.e. $\text{V}^{7/13}$ in F minor appears almost correctly as $[C, E, G\sharp, B\flat]$, i.e. as $\text{V}^{5\flat/7}$ (the difference is that in the first case the 13th interval was major).

d) Eight (8) chords were misspelled because they appeared over a pedal note (pedal notes were included in our GCT analysis, while they were omitted in Temperley’s analysis); e.g. $[D, A, C\sharp, G]$, a V^7 over a tonic pedal in D major, appeared as $[A, D, G, C\sharp]$, i.e. as $\text{V}^{4/7/10}$, and $[D, C\sharp, G, B]$, a vii^{07} over a tonic pedal, is described as $[G, B, D, C\sharp]$, i.e. as $\text{IV}^{11\sharp}$.

e) Two (2) sus4 chords $[0, 5, 7]$ were identified incorrectly as $[0, 5, 10]$; e.g. $[C, F, G]$, Vsus^4 in F major contains the dissonant interval $[F, G]$ and was erroneously re-ordered as $[G, C, F]$, i.e. as $\text{ii}^{4/7}$ (quartal chord).

On the other hand, the GCT algorithm correctly identified numerous functionally ambiguous chords, such as various cases of augmented 6th chords (mainly German types, but also Italian and French types) formed over a variety of scale degrees (6 \flat , 2 \flat , 4, etc.). It also correctly identified most harmonic circles of fifths, applied dominants, neapolitan chords, chords produced by modal mixture and complex triadic chords (with more than four members).

Overall, in the context of tonal music and the for standard tonal consonance vector, the GCT algorithm produces quite satisfactory results. However, it makes primarily the following types of mistakes: firstly, it yields ambiguous results regarding the root of symmetric chords such as the full diminished seventh and augmented chords – to disambiguate the root for symmetrical chords (mainly for diminished seventh chords), harmonic context has to be taken into account (e.g. the root of the following chord); secondly, it assigns the wrong root to chords that have ‘dissonant’ intervals at their triadic base, such as diminished fifths in half-diminished chords or major second in sus4 chords; thirdly, tertian chords that have notes missing from their base (e.g. missing third in seventh chords) are misinterpreted as their upper denser part is taken as the chords base and the lower root as an extension; and, finally, pedal notes, when taken into account for the identification of the GCT type, produce complex and functionally incorrect results.

In order to correct such cases, a more sophisticated model for harmonic analysis is required, which extends the purely representational scope of the current proposal. Such a model would take into account voicing (e.g. the bass note), chord transition probabilities (functions), and, even, higher-level domain-specific harmonic knowledge (e.g. specific types of chords used in particular idioms).

The GCT algorithm captures the common-practice roman-numeral harmonic analysis encoding scheme (for the ‘standard’ consonance vector) reasonably well. Additionally, it

¹ The dataset set is available in machine readable format at <http://theory.esm.rochester.edu/temperley/kp-stats/index.html>.

tonal: I IIN (V7) IIN V6/5 I vi ii6/5 V7 I
 [0, [0, 4, 7]] [1, [0, 4, 7]] [0, [0, 3, 6]] [1, [0, 4, 7]] [7, [0, 4, 7, 10]] [0, [0, 4, 7]] [9, [0, 3, 7]] [2, [0, 3, 7, 10]] [0, 4, 7, 10] [0, [0, 4, 7]]
 atonal: [5, [0, 4, 7]] [6, [0, 4, 7]] [5, [0, 3, 6]] [6, [0, 4, 7]] [10, [0, 2, 6, 9]] [5, [0, 4, 7]] [2, [0, 3, 7]] [5, [0, 2, 5, 9]] [10, [0, 2, 6, 9]] [5, [0, 4, 7]]

Figure 1. Beethoven, *Andante Favori*, reduction of mm.189-198. Tonal and atonal GCT analysis (see text).

adapts to non-tonal systems, such as atonal, octatonic or traditional polyphonic music. The question is whether the GCT representation works well on such non-tonal systems. The GCT representation has been employed in the case of traditional polyphonic music from Epirus [14], whereby, song transcriptions were initially converted to the GCT encoding, followed by a learning HMM scheme. This scheme was then employed to learn chord transitions, which was finally used to create new harmonisations in the polyphonic style of Epirus. Ongoing research is currently studying the application of GCT on various harmonic idioms, from medieval music to 20th century music, and various pop and folk traditions.

3. GROUPING GCT CHORDS

Chord relationships, and more specifically chord similarity/distance in tonal and non-tonal music, have been studied by various music theorists/researchers; some notable examples are the work by Hindemith [13], the classification scheme by Harris [11], pitch-class set (pcset) theory [8, 9], neo-riemannian theory [4, 5], tonal pitch space theory [19] and the work by Quinn [22]. Empirical studies have attempted to evaluate aspects of such theories in an empirical manner - see, for instance, [1, 16, 17, 24]. Apart from sensory, cognitive and musicological factors that play a significant role in such studies (and also in the first chord grouping algorithm below), the work herein makes additional use of data-driven information derived from statistical harmonic analysis in order to tackle similarity of different chord groups based on their functionality (i.e. transitions between chords) cf. related work by Quinn and Mavromatis [23].

A large number of unique note simultaneities may appear in a certain musical style. These simultaneities, however, are organised into fewer more cognitively manageable chord families/categories. Things like octave equivalence, interval inversion equivalence, root, tonal centre and so on, enable a parsimonious ‘packing’ of the great variety of actual note simultaneities into a relatively small number of musically meaningful chord categories. This categorical organisation of chords is probably most apparent in the case of tonal music; for instance, ‘major chord’ applies to many vertical note configurations that may appear in different guises such as open/closed position, different registers and keys, with doubled or missing or, even, extra notes.

The GCT algorithm re-organises note simultaneities in

terms of ‘root’, ‘base’, ‘extension’ and relative root to local key, giving the same label to pitch collections that have identical structure in relation to a tonal centre. However, missing or extra notes are not taken into account, resulting in a larger number of chords than what is musically acceptable (at least for tonal music). For instance, the GCTs: $[7, [0, 4, 7]]$, $[7, [0, 4]]$, $[7, [0, 4, 10]]$, $[7, [0, 4, 7, 10]]$ are all independent chord labels whereas they could be grouped under one dominant chord label (these share the same relative root and are all subsets of the $[0, 4, 7, 10]$ chord type). Additionally, the GCTs: $[11, [0, 3, 6]]$ and $[11, [0, 3, 6, 9]]$ are diminished chords on the seventh scale degree; these cannot be grouped with the previous GCTs because of the different relative root and chord type, even though we know that they also belong to the dominant chord functional category.

In the next two subsections, firstly, a simple algorithm is presented that groups raw GCTs into GCT chord categories based on GCT properties, such as, relative root, type similarity and relationship to underlying scale/key; secondly, an algorithm is developed that further organises the above GCT categories into functional chord categories by examining the function of chords, i.e., chords that tend to be followed by the same chords (similar rows in a chord transition matrix) are considered to have the same function. These two algorithms tidy up the initial raw GCTs into meaningful chord categories, each represented by the most frequently occurring instance (exemplar).

3.1 Grouping chords based on their GCT properties

Following the aforementioned example, the ‘exemplar $[7, [0, 4, 7]]$ ’ might be found in several ‘reduced’ (e.g. $[7, [0, 4]]$) or ‘expanded’ (e.g. $[7, [0, 4, 7, 11]]$) forms, that actually represent the same chord label. According to the GCT representation, further abstraction can be achieved through grouping GCT expressions of simultaneities that ‘evidently’ concern the same chord.

Grouping of GCTs has been studied under some basic assumptions about the chord characteristics that are reflected by the root scale degree, the base and the scale notes underlying a GCT expression. Specifically, GCT expressions are grouped into more general GCT categories that potentially contain several GCT members according to the criteria described below: two chords belong to the same group if

1. they have the same scale degree root,

2. their GCT bases are subset-related and
3. they both contain notes that either belong or not to the given scale context.

Regarding criterion 2, two bases B_1 and B_2 are considered subset-related if $B_1 \subseteq B_2$ or $B_2 \subseteq B_1$, e.g. $[0, 4] \subseteq [0, 4, 7]$ while $[0, 4] \not\subseteq [0, 3, 7]$. Criterion 3 is utilised to identify and group together chords that belong to secondary tonalities within the primary tonality of the piece. For instance, in a diatonic major context, while $c_1 = [0, [0, 4, 7]]$ and $c_2 = [0, [0, 4, 7, 10]]$ fulfil criteria 1 and 2, according to criterion 3 they are not grouped together since c_2 includes value 10, which is mapped to the non-diatonic 10 pitch class value. In a major context $[0, [0, 4, 7, 10]]$ is secondary dominant to the IV (V/IV) and is differentiated from the I major chord.

Each GCT group includes the GCT types that satisfy the aforementioned three criteria. Furthermore, each group is represented by the ‘exemplar’ GCT type, which is the one that is more often met in the datasets under study. Some common chord groups in the major scale Bach Chorales are illustrated in Table 2. This table also includes the functional naming of each group in order to assist the comparison of the derived GCT types and the standard roman-numeral labelling. Testing this simple algorithm on sets of both major and minor Bach chorales gives a reasonable first classification of the ‘raw’ GCTs.

3.2 Functional similarity of chords

According to functional harmony each chord can be viewed not only in terms of its actual pitches, roots, chord type and so on, but also in terms of its ‘dynamic’ attributes according to its position in a chord sequence and to the chords that usually follow [18]. For instance, in the tonal idiom, dominant chords are ‘expected’ to resolve to a (relative) tonic chord. Therefore, different chords can be similar according to the purpose they serve in terms of their functionality within chord sequences.

In this Section, a first approach to derive the functionality of the GCT chord groups is addressed by observing their succeeding chords in chord sequences extracted from specific idioms. In order to capture the functional relations between GCT groups of specific music idioms, the first-order Markov transition table is considered for all the GCT chord sequences that pertain to a certain idiom. The proposed approach below, tackles chord similarity by employing the Euclidean distance metrics related to the probability distribution for each chord group to precede any other (i.e., euclidean distance between rows of the transition matrix).

Figure 2 illustrates a colour-based graphic interpretation of the transition matrix obtained from a collection of Bach Chorales in major mode (darker areas indicate higher probabilities); transitions between chords that pertain to the same GCT chord group are disregarded (this neutralises the diagonal). Furthermore, GCT chord groups that occurred 4 times or less in the entire dataset were discarded, since their functional role can hardly be determined by so few observations. The probability that a GCT chord group is

followed by another (a row of the transition matrix in Figure 2) is regarded as a vector that defines the position of this group into the ‘space of transitions’. Thereby, functional relations between GCT groups according to their most common successors can be deduced by employing distance metrics between rows of the transition matrix.

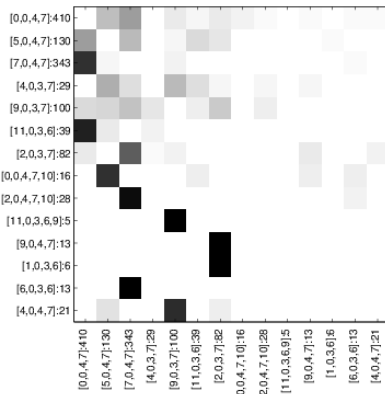


Figure 2. The first-order Markov transition matrix of GCT groups in the major Bach Chorales. The numbers after the colon indicate the number of times a representative of a GCT group was found in the data.

3.3 Functional similarity results

The Euclidean distance between transitions of GCT groups (rows in the transition matrix depicted in Figure 2) in a set of major Bach chorales has been utilised to produce the dendrogram of distances illustrated in Figure 3. For clarity of presentation, GCT groups with rare occurrences (less than 4) were not considered, although their placement in the grouping results was explainable. The six annotated clusters underpin interesting functional relations between the chords involved (the comments are presented in diminishing cluster coherence order):

Cluster 1 comprises the double dominant V/V $[2, [0, 4, 7, 10]]$ and its sub-tonic vii°/V $[6, [0, 3, 6]]$. Both chords have identical harmonic function (pre-dominant) and they always lead to the dominant V chord as applied dominants.

Cluster 4 contains the dominant V $[7, [0, 4, 7]]$ and the leading-tone triad vii° $[11, [0, 3, 6]]$, which is a subset of the dominant 7th chord. Both chords have strong dominant function.

Cluster 6 contains the applied dominant of the sub-dominant, i.e. V/vi , and the corresponding applied diminished 7th chord, i.e. vii°/vi . The GCT algorithm erroneously describes the second chord as its enharmonic equivalent $[11, [0, 3, 6, 9]]$, i.e. as $\text{vii}^{\circ 7}$ $[B, D, F, Ab]$, while it should be $[8, [0, 3, 6, 9]]$, i.e. $\text{vii}^{\circ 7}/vi$ $[G_#^+, B, D, F]$. However, the strong clustering relation could help to disambiguate the

functional name	exemplar	Group members			
tonic	$[0, [0, 4, 7]]$	$[0, [0, 4, 7]]$	$[0, [0, 4]]$	$[0, [0, 4, 7], [11]]$	
dominant	$[7, [0, 4, 7]]$	$[7, [0, 4, 7]]$	$[7, [0, 4, 7], [10]]$	$[7, [0, 4], [10]]$	$[7, [0, 4]]$
subdominant	$[5, [0, 4, 7]]$	$[5, [0, 4, 7]]$	$[5, [0, 4]]$	$[5, [0, 4, 7], [11]]$	
V/IV	$[0, [0, 4, 7], [10]]$	$[0, [0, 4, 7], [10]]$	$[0, [0, 4], [10]]$		

Table 2. Four tonal chord groups and their exemplar GCTs. Notice how the group of $[0, [0, 4, 7]]$ has been separated from the group of $[0, [0, 4, 7], [10]]$, due to the non-diatonic pitch class 10 of the latter.

root of the diminished 7th chord; this is future work for improving the descriptiveness efficiency of the GCT representation.

Cluster 5 groups the applied dominant of the supertonic, i.e. $V/ii[9, [0, 4, 7]]$, and the corresponding applied diminished triad, i.e. $vii^{\circ}/ii[1, [0, 3, 6]]$. Clusters 1, 4, 5, 6 are of the same category, as they share the same dominant function.

Cluster 2 is different, as it groups three chords that have (or may have) tonic harmonic function, the tonic I $[0, [0, 4, 7]]$, the submediant $vi[9, [0, 3, 7]]$ and the mediant $iii[4, [0, 3, 7]]$. In functional harmony [6], these chords are labeled as T, Tp and Tg accordingly and the last two chords have a diatonic (with two common tones) third-relation with the first.

Cluster 3 is similar to cluster 2, as it groups two chords with diatonic third-relation, however in this case the chords share subdominant harmonic function: the subdominant IV $[5, [0, 4, 7]]$ and the supertonic ii $[2, [0, 3, 7]]$. In functional harmony, they are described as S and Sp accordingly.

Overall, the proposed data-driven functional approach to chord grouping seems to be quite reliable. Further testing is necessary on larger and more varied corpora.

4. CONCLUSIONS

The paper at hand examines two main topics: a) the ability of the GCT algorithm to analyse chord sequences (in comparison to roman numeral analysis) and b) the possibility to organise the ‘raw’ GCT labels in higher-level chord families according to the internal GCT properties and to dynamic functional properties in terms of chord successions in harmonic corpora. The first study was based on comparing the annotations of chords produced by the GCT algorithm with the harmonic annotations of human experts (around 92% accuracy in the Kostka-Payne dataset). So, with its ability to identify roots and chord types, the GCT can be used as an interpretation/analytic tool allowing it to be classified as a hybrid between neutral representations (e.g. Forte pc-set theory analysis) and interpretative ones (e.g. roman numeral analysis). For the second study, information about transitions of the GCT chord groups were utilised to identify similarities between these groups according to their successors, thus, reflecting functional relations.

The results are promising, since they illustrate the ability of the GCT to accurately label chords, but also to reveal chord groups according to (higher) functional meaning in the tonal system. It is maintained that if the GCT representation can cope with such a sophisticated hierar-

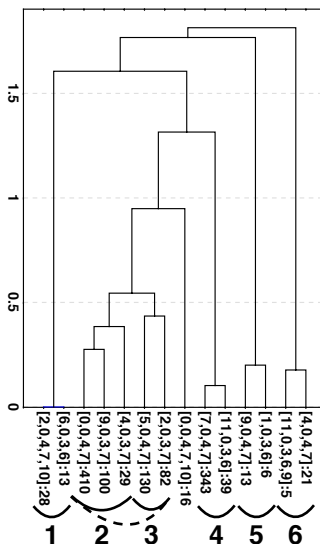


Figure 3. (a) The dendrogram derived from the Euclidean distances between rows of the transition matrix (Figure 2).

chic harmonic system as the tonal system, then it seems likely that it can deal with other non-tonal systems as well. Preliminary examples presented in [2, 3, 14] illustrate the potential of the GCT to represent non-tonal harmonic idioms; further research is under way to unveil the potential of the proposed representation in other musics.

5. ACKNOWLEDGEMENTS

This work is funded by the COINVENT project. The project COINVENT acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET-Open grant number: 611553.

6. REFERENCES

- [1] Emmanuel Bigand, Richard Parncutt, and Fred Lerdahl. Perception of musical tension in short chord se-

- quences: The influence of harmonic function, sensory dissonance, horizontal motion, and musical training. *Perception & Psychophysics*, 58(1):125–141, 1996.
- [2] E. Cambouropoulos. The Harmonic Musical Surface and Two Novel Chord Representation Schemes. In D. Meredith, editor, *Computational Music Analysis*, page (forthcoming). Springer, 2015.
- [3] Emiliós Cambouropoulos, Máximos Kaliakatsos-Papakostas, and Costas Tsougras. An idiom-independent representation of chords for computational music analysis and generation. In *Proceeding of ICMC–SMC 2014*, 2014.
- [4] Richard Cohn. Neo-riemannian operations, parsimonious trichords, and their “tonnetz” representations. *Journal of Music Theory*, pages 1–66, 1997.
- [5] Richard Cohn. Introduction to neo-riemannian theory: a survey and a historical perspective. *Journal of Music Theory*, pages 167–180, 1998.
- [6] Diether de la Motte. *Harmonielehre*. Kassel: Bärenreiter – Verlag, 1976.
- [7] Kemal Ebcioglu. An expert system for harmonizing four-part chorales. *Computer Music Journal*, 12(3):43–51, 1988.
- [8] Allen Forte. *The structure of atonal music*. Yale University Press, New Haven, 1973.
- [9] Allen Forte. Pitch-class set genera and the origin of modern harmonic species. *Journal of Music Theory*, pages 187–270, 1988.
- [10] Mark Thomas Granroth-Wilding. *Harmonic analysis of music using combinatory categorial grammar*. PhD thesis, Institute for Language, Cognition and Computation School of Informatics University of Edinburgh, Edinburgh, Scotland, November 2013.
- [11] Simon John Minshaw Harris. *A proposed classification of chords in early twentieth-century music*. PhD thesis, King’s College London (University of London), 1985.
- [12] Christopher Harte, Mark Sandler, Samer A. Abdallah, and Emilia Gómez. Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)*, pages 66–71, London, UK, 2005.
- [13] Paul Hindemith. *The craft of musical composition, vol.1: Theoretical part (Trans. A. Mendel)*. Associated music, New York, 1937/1942.
- [14] M. Kaliakatsos-Papakostas, A. Katsiavalos, C. Tsougras, and E. Cambouropoulos. Harmony in the polyphonic songs of epiros: Representation, statistical analysis and generation. In *4th International Workshop on Folk Music Analysis (FMA) 2014*, June 2014.
- [15] Máximos Kaliakatsos-Papakostas and Emiliós Cambouropoulos. Probabilistic harmonisation with fixed intermediate chord constraints. In *Proceeding of ICMC–SMC 2014*, 2014.
- [16] Carol L Krumhansl. *Cognitive foundations of musical pitch*, volume 17. New York: Oxford University Press, 1990.
- [17] Tuire Kuusi. Chord span and other chordal characteristics affecting connections between perceived closeness and set-class similarity. *Journal of New Music Research*, 34(3):259–271, 2005.
- [18] Steven G Laitz. *The complete musician: an integrated approach to tonal theory, analysis, and listening*. Oxford University Press, New York, 2012.
- [19] Fred Lerdahl. *Tonal pitch space*. Oxford University Press, 2001.
- [20] Francois Pachet and Pierre Roy. Musical harmonization with constraints: A survey. *Constraints*, 6(1):7–19, January 2001.
- [21] Jean-François Paiement, Douglas Eck, and Samy Bengio. Probabilistic melodic harmonization. In *Proceedings of the 19th International Conference on Advances in Artificial Intelligence: Canadian Society for Computational Studies of Intelligence, AI’06*, pages 218–229, Berlin, Heidelberg, 2006. Springer-Verlag.
- [22] Ian Quinn. Listening to similarity relations. *Perspectives of New Music*, 39, 2001.
- [23] Ian Quinn and Panayotis Mavromatis. Voice-leading prototypes and harmonic function in two chorale corpora. In *MCM*, pages 230–240. Springer, 2011.
- [24] Art G. Samplaski. *A comparison of perceived chord similarity and predictions of selected twentieth-century chord-classification schemes, using multidimensional scaling and cluster analysis*. PhD thesis, Indiana University, 2000.
- [25] M. Schorlemmer, A. Smaill, K.U. Kühnberger, O. Kutz, S. Colton, E. Cambouropoulos, and A. Pease. Coinvent: Towards a computational concept invention theory. In *5th International Conference on Computational Creativity (ICCC) 2014*, June 2014.
- [26] Ian Simon, Dan Morris, and Sumit Basu. Mysong: Automatic accompaniment generation for vocal melodies. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, CHI ’08*, pages 725–734, New York, NY, USA, 2008. ACM.
- [27] D. Temperley. Computational models of music cognition. In D. Deutsch, editor, *The psychology of music (2nd Edition)*. Academic Press, San Diego, 2012.

BEAT HISTOGRAM FEATURES FROM NMF-BASED NOVELTY FUNCTIONS FOR MUSIC CLASSIFICATION

Athanasios Lykartsis

Technische Universität Berlin
Audio Communication Group

alykartsis@mail.tu-berlin.de

Chih-Wei Wu

Georgia Institute of Technology
Center for Music Technology

cwu307@gatech.edu

Alexander Lerch

Georgia Institute of Technology
Center for Music Technology

alexander.lerch@gatech.edu

ABSTRACT

In this paper we present novel rhythm features derived from drum tracks extracted from polyphonic music and evaluate them in a genre classification task. Musical excerpts are analyzed using an optimized, partially fixed Non-Negative Matrix Factorization (NMF) method and beat histogram features are calculated on basis of the resulting activation functions for each one out of three drum tracks extracted (Hi-Hat, Snare Drum and Bass Drum). The features are evaluated on two widely used genre datasets (GTZAN and Ballroom) using standard classification methods, concerning the achieved overall classification accuracy. Furthermore, their suitability in distinguishing between rhythmically similar genres and the performance of the features resulting from individual activation functions is discussed. Results show that the presented NMF-based beat histogram features can provide comparable performance to other classification systems, while considering strictly drum patterns.

1. INTRODUCTION


The description of musical rhythm remains an important and challenging topic in Music Information Retrieval (MIR) with applications in several areas [12, 16]. The difficulty of rhythm extraction lies in its multifaceted character, which involves periodicity and structural patterning in the signal as well as perceptual components such as musical meter [19]. An approach which has achieved some popularity over the last years is based on the creation of a periodicity representation — commonly called the beat histogram (BH) — and the subsequent extraction of features from this histogram to be used, e.g., in genre classification [4, 13, 33]. A common first processing step of all approaches is the extraction of a so-called novelty function [2] or its derivatives as the starting point for further analysis. Since a complete rhythm representation of a musical track results from the superposition of the temporal progressions of different instruments or voices [12, 16], it makes sense to include features taking into account individual temporal and spectral properties.

In western popular music (which is the focus of this paper), rhythm is most often carried from the drum section, providing the temporal grid on which other instruments can unfold their melodic or harmonic patterns. This makes the analysis of the drum track appealing for the description of rhythmic character. In order to obtain the rhythmic properties of the drum section, the extraction of temporal novelty functions per instrument is necessary. Although such methods for the extraction of specific voices or instruments have been commonly used in the area of source separation or automatic instrument transcription (the most notable being non-negative matrix factorization (NMF) [31]), their application to rhythm extraction problems is, to the best of our knowledge, sparse. We therefore propose to use a technique for source separation and drum transcription based on partially fixed NMF using the resulting activation functions as a source material for the extraction of rhythmic features based on beat histograms. This paper investigates the suitability of the proposed features in the context of rhythm-based genre classification for dance music and other styles.

The paper is structured as follows. In the second section, an overview of previous work and the goals of the current paper are presented. In section 3, the drum transcription procedure and the feature extraction are described. In the fourth section, the evaluation of the proposed features and the results are given. After discussing the results in section 5, we close by giving conclusions and suggestions for future work (sect. 6).

2. PREVIOUS WORK AND GOALS

Beat histograms have been used for a long time as rhythmic descriptions. Initially introduced in studies on beat tracking and analysis [11, 29] as a useful very low frequency periodicity representation, they were only later referred to as the *beat histogram* [33] or *periodicity histogram* [13]. The histogram is useful as an intermediate representation that can be used to extract musical parameters such as tempo as well as low-level features (e.g., statistical properties of the histogram). Traditionally, a measure of the signal amplitude envelope or its change over time is utilized as the novelty function for the extraction of a beat histogram [4, 13, 33]. However, in the field of onset detection, the proposed novelty functions take into account spectral content changes [3, 10, 15, 27]. Genre classification systems based on such representations have generally shown

 © Athanasios Lykartsis, Chih-Wei Wu, Alexander Lerch. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Athanasios Lykartsis, Chih-Wei Wu, Alexander Lerch. "Beat histogram features from NMF-based novelty functions for music classification", 16th International Society for Music Information Retrieval Conference, 2015.

promising results, although rhythm features do usually not perform as well as features from other domains such as timbre descriptors [4,28,33]. However, studies have shown that for highly rhythmical music, beat histogram features can achieve very high performance [13], a fact which has been confirmed in current studies investigating the role of using multiple novelty functions as a basis for beat histogram features [20].

Since drum tracks convey essential information about tempo, rhythm and possibly genre, they could potentially provide better representation for extracting rhythm features. To extract drum tracks from complete mixtures of music, a drum transcription system for polyphonic music would be necessary. Gillet and Richard divide systems for the drum transcription from mixtures into three categories [9]: (i) segment and classify, (ii) separate and detect, and (iii) match and adapt. Here, we focus on the second type of approaches (separate and detect). Based on the assumption that the music signal is a superposition of different sound sources, the music content could be transcribed by first decomposing the signal into source templates with corresponding activation functions, and then detecting the activities of each template. Different methods such as Independent Subspace Analysis [7], Prior Subspace Analysis [6], and Non-negative Matrix Factorization [1,21] fall into this category. These approaches are usually easy to interpret since most of the decompositions result in spectrum-like representations. Furthermore, these approaches do not require additional classes for simultaneous events, which could potentially reduce the model complexity.

In the context of NMF for music transcription, the following issues have to be taken into consideration: First, the number of sound sources and notes within a music recording is usually unknown. It is therefore difficult to determine a suitable rank r in order to obtain a clear differentiation of the decomposed components in the dictionary matrix. Second, after the unsupervised NMF decomposition process, it is difficult to identify the associated instrument of each component in the dictionary matrix W when rank is too high or too low. Third, when multiple similar entries exist in the dictionary matrix, the corresponding activation matrix could be activated at these entries simultaneously, which in turn increases the difficulty of intuitively interpreting the results.

To address the above issues, Yoo et al. proposed a co-factorization algorithm [35] to simultaneously factorize a prior drum track and a target signal, and use the basis matrix from the drum track to identify the drum components in the target signal. This method ensures that the drum components in both dictionary matrices remain percussion only over the iterations, and thus proper isolation of the harmonic components from the drum components. Since they focus on drum separation rather than drum transcription, their selection of ranks can be higher, but the approach is not directly applicable to the transcription problem because of the probable lack of interpretability of the dictionary matrix. Wu and Lerch proposed a variant of the co-factorization algorithm using partially fixed NMF (PFNMF)

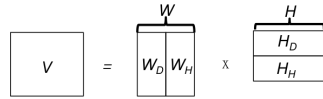


Figure 1. Illustration of the factorization process. W : dictionary matrix. H : activation matrix. Subscript D : drum components, Subscript H : harmonic components.

for drum transcription in polyphonic signals [34]. Instead of co-factorization, this method uses a pre-determined drum dictionary matrix during the decomposition process, and extracts one activation function for each of the three drums (Hi-Hat, Snare Drum, and Bass Drum).

In this paper, we apply PFNMF to transcribe drum events in polyphonic signals, and use the activation functions as the basis for the extraction of beat histogram features. The idea of using NMF with prior knowledge of targeting source within the mixture has been applied in source separation tasks [32], multi-pitch analysis [26] and drum transcription [34]. Furthermore, the use of multiple novelty functions for the extraction of beat histograms has been proposed in [20]. Here, we combine both approaches for the generation of rhythmic features which are descriptive of the percussive rhythmic content of polyphonic tracks and therefore of their general rhythmic character. We focus on two tasks: the investigation of their overall performance, in order to determine the salience of the features for genre classification; and their performance for each percussive component (drum track) separately, attempting to extract conclusions regarding the importance of drum based rhythm features and the salience of NMF activation functions.

3. METHOD

The basic concept of NMF is to approximate a matrix V with matrices W and H as $V \approx WH$ with non-negativity constraints. Given a $m \times n$ matrix V , NMF will decompose the matrix into the product of a $m \times r$ dictionary (or basis) matrix W and an $r \times n$ activation matrix H , with r being the rank of the NMF decomposition. In most audio applications, V is the spectrogram to be decomposed, W contains the magnitude spectra of the salient components, and H indicates the activation of these components with respect to time [31]. The matrices W and H are estimated through an iterative process that minimizes a distance measure between the target spectrogram V and its approximation [30].

To effectively extract drum activation functions from the polyphonic signals, PFNMF is used in this study. Figure 1 visualizes the basic concept from the work of Yoo et al.: the matrices W and H are split into the matrices W_D and W_H , and H_D and H_H , respectively. Instead of using co-factorization, PFNMF initializes the matrix W_D with drum components and to not modify it during the factorization process. Matrices W_H , H_H , and H_D are initialized with random numbers. The distance measure used in this paper is the generalized KL-divergence (or I-divergence), in which



Figure 2. Flowchart of NMF and beat histogram feature extraction and classification system.

$D_{KL}(x | y) = x \cdot \log(x/y) + (y - x)$. The cost function as shown in (1) is minimized by applying gradient descent and multiplicative update rules, the matrices W_H , H_H , and H_D will be updated according to Eqs. (2)–(4).

$$J = D_{KL}(V | W_D H_D + W_H H_H) \quad (1)$$

$$H_D \leftarrow H_D \frac{W_D^T (V / (W_D H_D + W_H H_H))}{W_D^T} \quad (2)$$

$$W_H \leftarrow W_H \frac{(V / (W_D H_D + W_H H_H)) H_H^T}{H_H^T} \quad (3)$$

$$H_H \leftarrow H_H \frac{W_H^T (V / (W_D H_D + W_H H_H))}{W_H^T} \quad (4)$$

PFNMF can be summarized in following steps:

1. Construct an $m \times r_D$ dictionary matrix W_D , with r_D being the number of drum components to be detected.
2. Given a pre-defined rank r_H , initialize an $m \times r_H$ matrix W_H , an $r_D \times n$ matrix H_D and an $r_H \times n$ matrix H_H .
3. Normalize W_D and W_H .
4. Update H_D , W_H , and H_H using (2)–(4).
5. Calculate the cost of the current iteration using (1).
6. Repeat step 3 to step 5 until convergence.

In our current setup, the STFT of the signals is calculated using a window size and a hop size of 2048 and 512, respectively. A pre-trained dictionary matrix is constructed from the training set, consisting of isolated drum sounds. The templates are extracted for the three classes Hi-Hat (HH), Bass Drum (BD) and Snare Drum (SD) as the median spectra of all individual events of one drum class in the training set. Next, the PFNMF will be performed with rank $r_H = 10$ on the test files. More details of the training process and the selection of rank r_H can be found in [34]. Finally, the activation Matrix H_D can be extracted from the audio signals through the decomposition process.

Once the activation functions of the three drum tracks have been extracted as described above, they are used as novelty functions for the calculations of beat histograms, similar to [20]. The complete procedure for the generation of a feature vector representing each track includes the following steps: For each activation function, the beat histogram is extracted through the calculation of an Auto-correlation Function (ACF) and the retaining of the area between 30 and 240 BPM. For each beat histogram, the sub-features listed in Table 1 are extracted. The concatenation

Distribution	Peak
Mean (ME)	Saliency of Strongest Peak (A1)
Standard Deviation (SD)	Saliency of 2nd Stronger Peak (A0)
Mean of Derivative (MD)	Period of Strongest Peak (P1)
SD of Derivative (SDD)	Period of 2nd Stronger Peak (P2)
Skewness (SK)	Period of Peak Centroid (P3)
Kurtosis (KU)	Ratio of A0 to A1 (RA)
Entropy (EN)	Sum (SU)
Geometrical Mean (GM)	Sum of Power (SP)
Centroid (CD)	
Flatness (FL)	
High Frequency Content (HFC)	

Table 1. Subfeatures extracted from beat histograms.

of all subfeature groups for each novelty function produces the final feature vector for an audio excerpt. Similar subfeatures as listed in Table 1 can be found in the literature, e.g., in [33] (Peak), and [4, 13] (Distribution). In total, 3 novelty functions are used for the production of as many beat histograms, from each of which 19 subfeatures are extracted, resulting in a total count of 57 features.

4. EVALUATION

4.1 Dataset Description

In order to evaluate the features for multiple track kinds possessing different rhythmic qualities, two datasets were considered: the Tzanetakis Dataset (GTZAN) [33], as an example of a dataset which is widely used, comprising 100 30 sec excerpts for each of 10 diverse musical genres; and the Ballroom Dataset [5, 13] (Ballroom), comprising 698 very rhythm/dance-oriented tracks of length 10 sec and therefore suitable for the evaluation of our NMF-based beat histogram features. Both datasets contain tracks with a drum section and others with only non-percussive instruments. This does not only allow to investigate if the extracted features are also suitable for music where a drum section is present and if they can generalize to other music styles, but also allows conclusions as to what genres in particular are represented satisfactory or insufficiently by the features.

4.2 Evaluation Procedure

The features were tested using the Support Vector Machine (SVM) algorithm for supervised classification. For our multiclass setting, an RBF kernel was used and the optimal parameters (C, γ) were determined through grid search. We chose the SVM classifier since it has been frequently used in similar genre classification experiments, shows generally good results (see [8]) and allows for comparability with those studies. Since the focus here lay on the features and not the classification algorithms, we refrained from using more state-of-the-art approaches such as deep learning algorithms. All experiments took place with a 10-fold cross-validation (using 90% of the data for training and 10% for testing over 10 randomly selected folds, taking the average accuracy over the folds for each dataset) and standardization (z-score) of the training and testing data. After the full

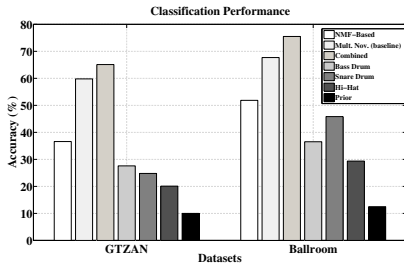


Figure 3. Classification results for both datasets.

NMF-based feature set (i.e., the features originating from all three drum activation functions) was tested, the features from each individual activation function were evaluated in turn in order to study the importance of each drum track separately. Finally, the NMF-based features are combined with other beat histogram features from a current study [20], extracted from novelty functions of amplitude (RMS), spectral shape (spectral flux, centroid, flatness and the first 13 MFCCs) and tonal components (pitch chroma coefficients and tonal power ratio) on 3 second-long frames. Those features resulted from a similar procedure as the one used here, where 30 different novelty functions were extracted and their beat histograms computed through the calculation of an ACF. A subsequent two-stage feature selection scheme (mutual information with target data [14] using the CMIM metric [25], followed by a sequential forward selection with an SVM wrapper [17]) was applied to retain the best-performing features, resulting in a total of 20 features in each case.

4.3 Results

The results are shown in Figure 3. On both datasets, the full NMF feature set (comprising features from all three drum activation functions) performs better than the individual ones (BD, SD, HH), with an attained accuracy of 36.6% and 51.9% for GTZAN and Ballroom, respectively. Those values lie considerably above the average priors of both datasets. The differences between the accuracies of the feature sets are not large (especially between the individual drum based feature sets) but are significant at the 0.05 level in all cases (based on a comparison test of the Cohen’s Kappa extracted from the confusion matrices). Due to their small values (ranging from 0.2% to 0.6%), standard deviations between accuracies of the folds for each feature set are not presented in Figure 3.

The multiple novelty feature set (from [20]) outperforms the NMF-based features, reaching an accuracy of 59.8% for the GTZAN and 67.7% for the Ballroom dataset, whereas the combined set (NMF and multiple novelty) demonstrates the best performance (accuracy of 65.1% (GTZAN) and 75.5% (Ballroom)). The individual feature sets from each drum track provide performance inferior to that of the

	Ch.	Ji.	Qu.	Ru.	Sa.	Ta.	Vw.	Wa.
Ch.	54	10	10	11	14	3	1	8
Ji.	17	13	5	6	10	2	2	5
Qu.	10	2	44	5	3	8	7	3
Ru.	8	3	2	53	4	7	2	19
Sa.	15	4	8	2	50	3	1	3
Ta.	2	1	6	6	2	55	7	7
Vw.	5	3	9	6	0	6	17	19
Wa.	5	0	4	16	1	4	4	76
Acc.	49	22	54	54	58	64	26	69
Pr.	15.9	8.6	11.7	14.0	12.3	12.3	9.3	15.8

Table 2. Confusion matrix for Ballroom dataset, average accuracy: 51.9%. Accuracy and Prior are given in %.

	Bl.	Cl.	Co.	Di.	Hi.	Ja.	Me.	Po.	Re.	Ro.
Bl.	15	11	16	15	4	7	9	9	11	3
Cl.	4	63	3	1	1	14	5	3	1	5
Co.	6	6	38	12	4	5	6	6	11	6
Di.	13	1	6	43	6	1	8	5	12	5
Hi.	8	4	5	4	21	8	10	20	13	7
Ja.	8	17	5	0	7	38	9	7	7	2
Me.	3	11	7	7	2	6	51	2	2	9
Po.	7	6	6	5	14	5	5	33	12	7
Re.	6	3	6	6	6	4	1	11	53	4
Ro.	6	4	10	10	17	10	11	11	10	11
Acc.	15	63	38	43	21	38	51	33	53	11
Pr.	10	10	10	10	10	10	10	10	10	10

Table 3. Confusion matrix for GTZAN dataset, average accuracy: 36.6%. Accuracy and Prior are given in %.

full NMF-based set, but still considerably higher than the prior. The best individual drums are the BD and SD for the GTZAN and Ballroom datasets, respectively. The worst individual percussion instrument is in both cases the HH. For the full NMF-based feature set, confusion matrices resulting from the classification can be seen in Tables 2 and 3. In general, features achieved better average performance on the Ballroom dataset than on the GTZAN. In order to evaluate the misclassifications and the performance of the individual genres, a closer observation of the confusion matrices of each dataset should be taken.

For the Ballroom dataset, confusions between genres appear to be plausible based on what one would expect when extracting rhythm features only from drums tracks: genres with strongly pronounced, stable rhythm played from a drum section such as samba and chachacha (Ch.) are confused with each other, whereas the waltz (Wa.) and tango (Ta.) genres, having no drum section (but still a succinct rhythm) are not confused much with other genres. The latter are the two genres which also achieve the best individual performance, followed by chachacha, quickstep (Qu.), rumba (Ru.) and samba (Sa.). Jive (Ji.) and viennese waltz (Vw.) display the worse performance, and are confused with chachacha and waltz respectively, a result which is also expected when one considers the rhythmic proximity of those genres, whether they possess a drum section or not.

For the GTZAN dataset, misclassifications present a more mixed picture: On the one hand, genres which possess tracks featuring a well articulated, distinct rhythmic performed by a drum section (such as reggae (Re.), metal

(*Me.*) and *disco* (*Di.*)) as well as the only genre without drums (*classical* (*Cl.*)) achieve satisfactory performance and are confused with genres which are rhythmically relatively close (*classical* with *jazz* (*Ja.*), *metal* with *rock* (*Ro.*), *disco* with *reggae*, and *reggae* with *pop* (*Po.*)). On the other hand, genres possessing tracks with a more “generic” rhythm (such as *country* (*Co.*) and *pop*) are confused with multiple other genres. Finally, *hiphop* (*Hi.*), *blues* (*Bl.*) and *rock* attain the last places in individual performance and are confused with multiple other genres.

5. DISCUSSION

The results show that beat histogram features based on NMF activation functions of specific drums can be helpful in rhythm-based genre classification, as their accuracy for the used datasets is comparable to that achieved by other rhythmic feature sets used up to date (59.8% [20] and 28% [33] for the GTZAN, 67.7% [20] and 56.7% [13] for the Ballroom dataset). When taking into account that the features are solely based on drum novelty functions, their performance, especially for the Ballroom dataset, can be seen as satisfactory. It is clear, though, that for this reason, our results cannot achieve as high accuracy as other studies which use very sophisticated methods [8, 18, 22–24]. Our results are somewhat lower than the state of the art using rhythm [22, 24] or combined features [8, 23], however staying in the same range. For the sake of comparison, we report here the highest performances reached when using advanced rhythmic features: on the GTZAN dataset an accuracy of 92.4% [22] has been achieved, for the Ballroom dataset one of 96.1% [24]. The advantage of our proposed methods and features lies in the ability to pinpoint the importance of the rhythm patterns from specific drums for specific genres.

The misclassifications (reported in Tables 2 and 3) show that genres which do not feature genre-specific rhythm patterns, even if those are clearly articulated by the drum section (e.g., a 4/4 BD and SD alternating beat), tend to be confused with other similar genres (especially when drum tracks are present, such as in *rock*). Genres containing non-percussive tracks (such as *classical* and *waltz*) or very specific rhythmic patterns (*reggae*) are more easily distinguished from others. Those results indicate that the NMF-based beat histogram features indeed capture rhythmic properties related to the drum section and the regularities of their periodicities, pointing towards the suitability of those features for the extraction of drum-based rhythmic properties and the use in the discrimination of musical tracks which contain drums from ones which do not.

With regards to the feature sets, the satisfactory accuracy of the NMF-based feature set is a hint towards the appropriateness of the features for the analysis of the rhythmic character of a musical track. However, it is clear that those features, being derived only from drum tracks, cannot represent as much information as features resulting from the use of multiple novelty functions covering many aspects of the signal temporal progress. The improved performance of the combined set (NMF and multiple novelty based)

is a consequence of incorporating specific, drum-related rhythm information in the feature base, showing that the NMF-based rhythm feature set can contribute information not provided by more general rhythm features and lead to significant improvement for the two evaluated datasets. The analysis of the features derived from the activation function of a specific drum track showed that mainly the snare drum and to a lesser extent the kick drum are the most important components. The tendency is strong for the Ballroom dataset, where the SD outperforms the BD, whereas for the GTZAN dataset the result is reversed but with a smaller difference. In all cases (also between the individual drum sets), the differences in accuracies between the feature sets are significant at the 5% level. Those results can be due to the very pronounced sound texture and greater power of those drums which leads to a salient activation function, as well as their role in providing the basic metric positions in most of western popular music. However, the accuracy of each subset lies below that of their combination, leading to the conclusion that the activation functions of all three percussion instruments contribute valuable information to the feature description of musical genre.

Concerning the datasets, the poorer classification performance observed for the GTZAN dataset is a sign of the more diverse character of tracks and genres in this set, containing music styles which lack a specific rhythmic character and can therefore not be distinguished effectively through beat histogram features derived from drum activation functions. Results were still better than the ones reported in [33], but their inferiority compared to the ones in other studies [13, 20] shows that when considering a multitude of different genres, solely drum based activation functions can not provide a complete rhythmic characterization. This, however, points towards the possible goal of using NMF in order to transcribe not only drums but also other instruments in order to use their activation functions as a basis for beat histogram features. The Ballroom dataset shows better performance, which was to be expected since the tracks therein are selected for belonging to different dance styles, requiring a special rhythmic pattern which is mostly conveyed by the drum section. The results are in the same range as those provided in [13] (56.7%) when using only periodicity histogram features. Furthermore, in the same study it was shown that using the tempo of the given tracks as a feature they could achieve very high results using a simple 1-NN classifier (51.7% for the “naive tempo” derived from the periodicity histogram and 82.3% for the ground-truth tempo provided with the recordings), reaching as much as 90% when combining the correct tempo with other descriptors (MFCCs) from the periodicity histogram. This shows that beat histograms (from which the tempo can be extracted) are a good tool for rhythmic analysis in datasets containing dance music such as the Ballroom.

Regarding specific genres, it is clear from the results that the NMF-based features have a twofold use: first, in representing genres which are characterized by distinct patterns in their drum sections (e.g., *reggae* or *samba*) and second, in characterizing genres which lack a drum section

at all (*waltz, classical*) in contrast to genres which do; the activation functions transcribed in this case are maximally different, leading to beat histogram features which can be easily discriminated by a classifier. Such a finding shows that drum-based rhythm features can be very helpful for rhythmic characterization of specific genres, which could be an argument for their further application when a specific kind of music is involved. As a general remark, it can be seen that genres possessing a stable rhythm articulated by a drum section such as *reggae* and *samba* or genres lacking drums in general (*waltz* and *classical*) perform better, whereas genres which have a very uncharacteristic rhythm (such as *rock* or *blues*) get more easily confused.

6. CONCLUSIONS

The work presented in this paper focuses on the creation of novel, NMF-based beat histogram features for rhythm-based musical genre classification and rhythmic similarity. The difference in comparison to other well-known studies for rhythm features based on beat histograms [4, 13, 24, 33] is the use of the activity functions of specific drums provided through NMF as a basis for the calculation of the beat histogram. We showed that the classification accuracy using these beat histogram features is comparable to that of other rhythm features, whereas our proposed features are better especially for characterizing tracks with specific rhythmic patterns or for distinguishing between songs with and without a drum section. It was observed that the most important percussion patterns for dance music classification were generated by the snare and the kick drum, which underlines the importance of its activation function for further tasks.

One future goal is the expansion of the use of NMF to identify more instruments or voices and use them as possible novelty functions. The goal would be to therefore capture the rhythmic patterns of every instrument, essentially joining source transcription and rhythm feature extraction into one module. Another possibility is the use of our proposed features for larger and more specific datasets, in order to further investigate their suitability for specific genres, as well as the strengths and weaknesses of the patterns extracted from individual drums in discriminating between musical genres. As an expansion of the feature selection procedure, a further idea would be to profit from the combination of NMF-based features and other acoustic features using a classifier that is capable of learning feature importance (e.g. random forest) to quantitatively investigate the importance of NMF-derived features. While NMF-based beat histogram features have been evaluated only in the context of rhythmic genre classification, we believe that they can prove useful in other tasks. Future research will focus on adjusting and using the proposed features for MIR tasks such as rhythmic similarity computation and structural analysis.

7. REFERENCES

- [1] David S Alves, Jouni Paulus, and José Fonseca. Drum transcription from multichannel recordings with non-negative matrix factorization. In *Proceedings of the European Signal Processing Conference (EUSIPCO), Glasgow, Scotland, UK, 2009*.
- [2] Juan P. Bello, Laurent Daudet, Samer Abdallah, Chris Duxbury, Mike Davies, and Mark B Sandler. A tutorial on onset detection in music signals. *IEEE Transactions on Speech and Audio Processing*, 13(5):1035–1047, 2005.
- [3] Juan P. Bello, Chris Duxbury, Mike Davies, and Mark Sandler. On the use of phase and energy for musical onset detection in the complex domain. *IEEE Signal Processing Letters*, 11(6):553–556, 2004.
- [4] Juan José Burred and Alexander Lerch. A hierarchical approach to automatic musical genre classification. In *Proceedings of the 6th international conference on digital audio effects*, pages 8–11, 2003.
- [5] Simon Dixon, Elias Pampalk, and Gerhard Widmer. Classification of dance music by periodicity patterns. In *Proceedings of the 4th International Conference on Music Information Retrieval (ISMIR)*, 2003.
- [6] Derry FitzGerald, Bob Lawlor, and Eugene Coyle. Drum transcription in the presence of pitched instruments using prior subspace analysis. In *Proceedings of the Irish Signals and Systems Conference (ISSC)*, 2003.
- [7] Derry FitzGerald, Robert Lawlor, and Eugene Coyle. Sub-band independent subspace analysis for drum transcription. In *Proceedings of the Digital Audio Effects Conference (DAFX)*, pages 65–59, 2002.
- [8] Zhouyu Fu, Guojun Lu, Kai Ming Ting, and Dengsheng Zhang. A survey of audio-based music classification and annotation. *IEEE Transactions on Multimedia*, 13(2):303–319, 2011.
- [9] Olivier Gillet and Gaël Richard. Transcription and separation of drum signals from polyphonic music. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(3):529–540, 2008.
- [10] Masataka Goto and Yoichi Muraoka. Music understanding at the beat level – real-time beat tracking for audio signals. In *Computational auditory scene analysis*, pages 157–176, August 1995.
- [11] Masataka Goto and Yoichi Muraoka. A real-time beat tracking system for audio signals. In *Proceedings of the International Computer Music Conference (ICMC)*, pages 171–174, 1995.
- [12] Fabien Gouyon and Simon Dixon. A review of automatic rhythm description systems. *Computer Music Journal*, 29(1):34–35, 2005.

- [13] Fabien Gouyon, Simon Dixon, Elias Pampalk, and Gerhard Widmer. Evaluating rhythmic descriptors for musical genre classification. In *Proceedings of the AES 25th International Conference*, pages 196–204, 2004.
- [14] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, 3:1157–1182, 2003.
- [15] Stephen Hainsworth and Malcolm Macleod. Onset detection in musical audio signals. In *Proceedings of the International Computer Music Conference (ICMC)*, 2003.
- [16] Enric Guaus i Termens. New approaches for rhythmic description of audio signals. Technical report, Music Technology Group, Universitat Pompeu Fabra, 2004.
- [17] Ron Kohavi and George H John. Wrappers for feature subset selection. *Artificial intelligence*, 97(1):273–324, 1997.
- [18] Chang-Hsing Lee, Jau-Ling Shih, Kun-Ming Yu, and Hwai-San Lin. Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features. *Multimedia, IEEE Transactions on*, 11(4):670–682, 2009.
- [19] Justin London. *Hearing in time*. Oxford University Press, 2012.
- [20] Athanasios Lykartsis. Evaluation of accent-based rhythmic descriptors for genre classification of musical signals. Master’s thesis, Audio Communication Group, Technische Universität Berlin, (www.ak.tu-berlin.de/menue/abschlussarbeiten/), 2014.
- [21] Arnaud Moreau and Arthur Flexer. Drum transcription in polyphonic music using non-negative matrix factorization. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, pages 353–354, 2007.
- [22] Yannis Panagakis, Constantine Kotropoulos, and Gonzalo R Arce. Music genre classification using locality preserving non-negative tensor factorization and sparse representations. In *Proceedings of the 10th International Conference on Music Information Retrieval (ISMIR)*, 2009.
- [23] Yannis Panagakis, Constantine L Kotropoulos, and Gonzalo R Arce. Music genre classification via joint sparse low-rank representation of audio features. *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, 22(12):1905–1917, 2014.
- [24] Geoffroy Peeters. Spectral and temporal periodicity representations of rhythm for the automatic classification of music audio signal. *IEEE Transactions on Audio, Speech and Language Processing*, 19(5):1242–1252, 2011.
- [25] Hanchuan Peng, Fulmi Long, and Chris Ding. Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.
- [26] Stanisław A Raczynski, Nobutaka Ono, and Shigeki Sagayama. Multipitch analysis with harmonic nonnegative matrix approximation. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR)*, 2007.
- [27] Axel Roebel. Onset detection in polyphonic signals by means of transient peak classification. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, 2005.
- [28] Nicolas Scaringella, Giorgio Zoia, and Daniel Mlynek. Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine*, 23(2):133–141, 2006.
- [29] Eric D Scheirer. Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, 103(1):588–601, 1998.
- [30] D Seung and L Lee. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.
- [31] Paris Smaragdis and Judith C Brown. Non-negative matrix factorization for polyphonic music transcription. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics.*, pages 177–180, 2003.
- [32] Paris Smaragdis, Bhiksha Raj, and Madhusudana Shashanka. Supervised and semi-supervised separation of sounds from single-channel mixtures. In *Independent Component Analysis and Signal Separation*, pages 414–421. Springer, 2007.
- [33] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE transactions on Speech and Audio Processing*, 10(5):293–302, 2002.
- [34] Chih-Wei Wu and Alexander Lerch. Drum transcription using partially fixed non-negative matrix factorization. In *Proceedings of the European Signal Processing Conference (EUSIPCO)*, 2015.
- [35] Jiho Yoo, Minje Kim, Kyeongok Kang, and Seungjin Choi. Nonnegative matrix partial co-factorization for drum source separation. In *Proceedings of the IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 1942–1945, 2010.

MUSIC SHAPELETS FOR FAST COVER SONG RECOGNITION

Diego F. Silva Vinícius M. A. Souza Gustavo E. A. P. A. Batista
Instituto de Ciências Matemáticas e de Computação – Universidade de São Paulo

{diegofsilva, vsouza, gbatista}@icmc.usp.br

ABSTRACT

A cover song is a new performance or recording of a previously recorded music by an artist other than the original one. The automatic identification of cover songs is useful for a wide range of tasks, from fans looking for new versions of their favorite songs to organizations involved in licensing copyrighted songs. This is a difficult task given that a cover may differ from the original song in key, timbre, tempo, structure, arrangement and even language of the vocals. Cover song identification has attracted some attention recently. However, most of the state-of-the-art approaches are based on similarity search, which involves a large number of similarity computations to retrieve potential cover versions for a query recording. In this paper, we adapt the idea of time series shapelets for content-based music retrieval. Our proposal adds a training phase that finds small excerpts of feature vectors that best describe each song. We demonstrate that we can use such small segments to identify cover songs with higher identification rates and more than one order of magnitude faster than methods that use features to describe the whole music.

1. INTRODUCTION

Recording or live performing songs previously recorded by other composers are typical ways found by several early-career and independent musicians to publicize their work. Established artists also play versions composed by other musicians as a way to honor their idols or friends, among other reasons. These versions of an original composition are popularly called *cover songs*.

The identification of cover songs has different uses. For instance, it can be used for estimating the popularity of an artist or composition, since a highly covered song or artist is an indicative of the popularity/quality of the composition or the author's prestige in the musical world. In a different scenario, a search engine for cover songs can help music consumers to identify different versions of their favorite songs played by other artists in different music styles or language.

Musicians that upload cover versions to websites such as *YouTube*, *Last.fm* or *SoundCloud* frequently neglect that the original songs may be copyright-protected. Copyright is a legal right created by the law that grants the creator of an original work (temporary) exclusive rights to its use and distribution. Legally speaking, when an interpreter does not possess a license to distribute his/her recording, this version is considered illegal.

For these reasons, cover song recognition algorithms are essential in different practical applications. However, as noted by [12], the automatic identification of cover songs is a difficult task given that a cover may differ from the original song in key, timbre, tempo, structure, arrangement and language of the vocals.

Another difficulty faced by automatic cover song identification systems, particularly those based on expensive similarity comparisons, is the time spent to retrieve recordings that are potential covers. For instance, websites such as *YouTube* have 300 hours of video (and audio) uploaded every minute¹. A significant amount of these videos is related to music content. Therefore, cover song identification algorithms have to be efficient in terms of query processing time in order to handle such massive amounts of data.

This paper proposes a novel algorithm to efficiently retrieve cover songs based on small but representative excerpts of music. Our main hypothesis is that we can characterize a specific music with small segments and use such information to search for cover songs without the need to check the whole songs.

Our hypothesis is supported by the success of a similar technique used in time series classification, named *shapelets* [16]. Informally, shapelets are time series subsequences, which are in some sense maximally representative of a class. For time series, shapelets provide interpretable and accurate results and are significantly faster than existing approaches.

In this paper, we adapt the general idea of shapelets for content-based music retrieval. For this, we evaluate several different ways to adapt the original idea to music signals. In summary, the main contributions of our proposal are:

- Our method adds a training phase to the task of content-based music information retrieval, which seeks to find small excerpts of feature vectors that best describe each signal. In this way, we make the similarity search faster;



© Diego F. Silva, Vinícius M. A. Souza, Gustavo E. A. P. A. Batista. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Diego F. Silva, Vinícius M. A. Souza, Gustavo E. A. P. A. Batista. "Music Shapelets for Fast Cover Song Recognition", 16th International Society for Music Information Retrieval Conference, 2015.

¹ www.youtube.com/yt/press/statistics.html.

- Even with small segments, we demonstrate that we can improve the identification rates obtained by methods that use features to describe the whole music;
- We show how to use our proposal along with a specific retrieval system. However, we note that our method can be added to any algorithm based on a similar sequence of steps, even methods to further speed-up the query. To do this, we simply need to apply such an algorithm on the shapelets, instead of the complete features vectors.

2. BACKGROUND AND RELATED WORK

The task of cover song recognition can be described as the following: given a set, S , of music recordings and a query music, q , we aim to identify if q is a version of one of the songs in S . Thus, a cover song recognition system can be considered a querying and retrieval system.

The state-of-the-art querying and retrieval systems can be divided into five main blocks [12]: *i*) feature extraction; *ii*) key invariance; *iii*) tempo invariance; *iv*) structure invariance; and *v*) distance calculation. Figure 1 illustrates these steps. This general framework leaves open which method will be applied in each step.

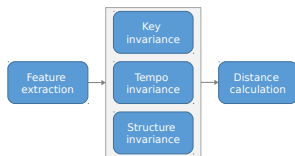


Figure 1. General retrieval system blocks. The feature extraction and distance calculation are required and should appear in this order. The other ones may provide best results, but are optional

Feature extraction is a change of representation from the high-dimensional raw signal to a more informative and lower-dimensional set of features. Chroma-features or pitch class profiles (PCP) are among the most used features for computing music similarity. These features are a representation of the spectral energy in the frequency range of each one of the twelve semitones. A good review of PCP, as well as other chroma-based features, can be found in [7].

Transpose a music for another key or main tonality is a commonly used practice to adapt the song to a singer or to make it heavier or lighter. Key invariance tries to reduce the effects of these changes in music retrieval systems that use tonal information. A simple and effective method to provide robustness to key changes is the optimal transposition index (OTI) [11]. As a first step, this method computes a vector of harmonic pitch class profiles (HPCP) for each song, which is the normalized mean value of the energy in each semitone [5]. When comparing two songs A and B , the method fixes the HPCP of A . For each shift of the

HPCP of B , it measures the inner product between the two vectors. The shift that maximizes this product is chosen and the song B is transposed using such a shift value.

Tempo invariance is the robustness to changes between different versions caused by faster or slower performances. One way of achieving tempo invariance is by modifying the feature extraction phase to extract one or more feature vectors per beat [4], instead of a time-based window. Another possibility is the use of specific feature sets, such as chroma energy normalized statistics (CENS) [8]. These features use a second stage in the chroma vector estimation that provides a higher robustness to local tempo variations.

Structure invariance is the robustness to deviations in long-term structure, such as repeated chorus or skipped verses. This invariance may be achieved by several different approaches, such as dynamic programming-based algorithms [3], sequential windowing [13] or by summarizing the music pieces into their most repeated parts [9].

The last step of a querying and retrieval system is the similarity computation between the query and reference data by means of a distance calculation. The most common approaches for this task are dynamic programming based algorithms that try to find an optimal alignment of feature vectors. A well-known example of this approach is the Dynamic Time Warping (DTW) distance function.

In this paper, we present an approach that adds a training phase to this process. This step seeks to find the most significant excerpt of each song in the set S (training set). These small segments are used in a comparison with the query song q . Our method is inspired by the idea of time series shapelets, presented next.

3. SHAPELETS

Time series shapelets is a well-known approach for time series classification [16]. In classification, there exists a training set of labeled instances, S . A typical learning system uses the information in S to create a classification model, in a step known as training phase. When a new instance is available, the classification algorithm associates it to one of the classes in S .

A time series shapelet may be informally defined as the subsequence that is the most representative of a class. The original algorithm of [16] finds a set of shapelets and use them to construct a decision tree classification model. The training phase of such learning system consists of three basic steps:

- **Generate candidates:** this step consists in extracting all subsequences from each training time series;
- **Candidates' quality assessment:** this step assesses the quality of each subsequence candidate considering its class separability;
- **Classification model generation:** this step induces a decision tree. The decision in each node is based on the distance between the query time series and a shapelet associated to that node.

In the first step, the length of the candidates is an intrinsic parameter of the candidates generation. The original algorithm limits the search to a range between a minimum (min_{len}) and maximum (max_{len}) length. All the subsequences with length between min_{len} and max_{len} are stored as candidates.

Given a candidate s , we need to measure the distance between s and a whole time series x . Notice that a direct comparison between them is not always possible since s and x can have very different lengths. Consider l as the candidate's length. The $distance(s, x)$ is defined as the smallest Euclidean distance between the candidate s and each subsequence of x with l observations.

The next steps of the shapelet algorithm are directly related to the classification task. Since this is not our focus, we suppress details of the algorithm from this point.

The general idea of classifying time series by shapelets is to use the distances between candidates and training time series to construct a classification model. First, the algorithm estimates the best information gain (IG) that can be obtained by each candidate. This is made by grouping the training examples that are closer – according a distance threshold – from the training examples that are more distant from the candidate. The best value for the threshold – called *best split point* – is defined by assessing the separation obtained by different values.

Finally, the algorithm uses the IG to create a decision tree. A decision node uses the information of the best shapelet candidate. In order to decide the class of a test example, we measure the distance between the query and the shapelet. If the distance is smaller or equal to the split point, its class is the one associated with the shapelet. Otherwise, the query is labeled as belonging to the other class.

For details on how to find the optimal split point and the decision tree's construction, we refer the reader to [16].

4. OUR PROPOSAL: MUSIC SHAPELETS

In this paper, we propose to adapt the idea of shapelets for a fast content-based music retrieval, more specifically for cover songs identification. Our adaptations are detailed in the next sections.

4.1 Windowing

The original approach to finding subsequence candidates uses sliding windows with different lengths. These lengths are the enumeration of all values in a range provided by the user. The sliding window swipes across the entire time series and such a process is performed for each example in the training set. We found this process to be very time consuming, accounting for most of the time spent in the training phase.

We note that music datasets are typically higher-dimensional than most time series benchmark datasets, in both number of objects as well as number of observations. Thus, we use a reduced set of specific values as window length instead of an interval of values. We empirically

noted that it is possible to find good candidates without enumerating all the lengths in a given range.

In addition, the original approach uses a sliding window that starts at every single observation of a time series. We slightly modified it so that the sliding windows skip a certain amount of observations proportional to the window length. This windowing technique with partial overlapping is common in audio analysis.

4.2 Dynamic Time Warping

Shapelets use Euclidean distance (ED) as the similarity measure to compare a shapelet and a whole time series. However, ED is sensitive to local distortions in the time axis, called *warping*. Warping invariance is usually beneficial for music similarity due to the differences in tempo or rhythm that can occur when a song is played live or by different artists.

In order to investigate this assumption, we evaluate the use of ED and Dynamic Time Warping (DTW) to compare shapelets extracted from music data. There is an obvious problem with the use of DTW, related to its complexity. While ED is linear on the number of observations, DTW has a quadratic complexity. Nevertheless, there is a plethora of methods that can be used so that we may accelerate the calculation of the distance between a shapelet and a whole music [10].

4.3 Distance-based Shapelet Quality

Shapelets were originally proposed for time series classification. In cover song identification we are interested in providing a ranking of recordings considering the similarity to a query. Therefore, IG is not the best choice to measure the candidates' quality.

IG in shapelet context finds the best split points and candidates according to class separability. However, music retrieval problems typically have a large number of classes (each class representing a single song) with few examples (different recordings of a certain song), hindering the analysis of class separability.

For this reason, we propose and evaluate the substitution of the IG by a distance-based criterion. We consider that a good candidate has a small distance value to all the versions of the related song and a high distance value to any recording of another song. Thus, we propose the criterion *DistDiff*, defined in Equation 1.

$$DistDiff(s) = \min_{i=1..n} (distance(s, OtherClass(i))) - \frac{1}{m} \sum_{i=1}^m distance(s, SameClass(i)) \quad (1)$$

where s is a candidate for shapelet, *SameClass* is the set of m versions of the song from were the candidate come from, *OtherClass* is the set of n recordings that does not represent a version of the same composition than the origin of s and $distance(s, Set(i))$ is the distance between the

candidate and the i -th recording in *Set* (*SameClass* or *OtherClass*).

Clearly, we are interested in candidates that provide a high value to the first term and a small value to the second. So, as higher the value of *DistDiff*, higher the quality of the candidate. In case of draw, we use the minimum average rank of the versions of the song related to s as tie breaking. In other words, if two candidates have the same value of *DistDiff*, the best candidate is the one that provides the best average ranking positions for the versions of the song from where s comes from.

4.4 Similarity

Since the technique of time series shapelets is interested in class separability, it stores at most one shapelet per class. On the other hand, in our problem we are interested in all examples of each "class label". So, we store one shapelet per recording in the training set, instead one for each composition.

The final step, the querying and retrieval itself, is made in two simple steps. First, our method measures the distance between the query music and each of the shapelets found in the training phase. Finally, the ranking is given by sorting these distances in ascending order.

4.5 Triplets

In a real scenario where the task of music retrieval will be performed, it is highly probable that a specific song has one to three authorized versions such as the original recording in a studio, an acoustic and a live version. Obviously, there are exceptions such as remix and many versions of live performances. Thus, when we extract shapelets from these songs in a conventional way, we have only a few instances for each class in the training set. This may hamper the candidate's quality calculation.

In addition, only a small segment of a song can be uninformative. This fact has been observed in other application domains. For instance, [14] uses features from the beginning, the middle and the end of each recording to perform the genre recognition task.

For these reasons, we also evaluated the idea of representing each recording as three shapelets. Figure 2 illustrate this procedure. The first step of this procedure divides the feature vector into three parts of the same length. After that, we find the most representative subsequence of each segment. Finally, during the retrieval phase, we use the mean distance from a query recording to each of the three shapelets. We will refer to these triple of shapelets as *triplets*.

5. EXPERIMENTAL EVALUATION

In this section, we present the datasets used in our evaluation and the experimental results. We conclude this section discussing the advantages of our method in terms of time complexity in the retrieval phase.

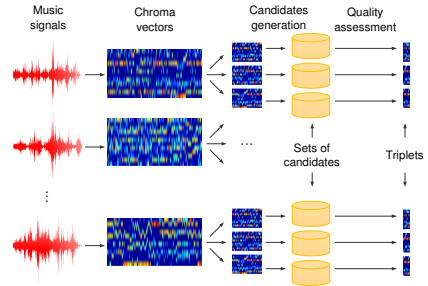


Figure 2. General procedure to generate triplets

5.1 Datasets

We evaluate our proposal in two datasets with different music styles. The first dataset is composed by classical music while the second contains popular songs.

The dataset *123 Classical* was originally used in [1]. This dataset has 123 different recordings concerning 19 compositions from Classical (between 1730 and 1820) and Romantic (between 1780 and 1910) ages. From the 123 recordings, 67 were performed by orchestras and the remaining 56 were played in piano.

We also collected popular songs from videos of *YouTube* and built a dataset named *YouTube Covers*. We made the *YouTube Covers* dataset freely available in our website [15] for interested researchers. This dataset was built with the goal of evaluating our proposal in a more diverse data since the covers songs in the *123 Classical* dataset in general faithfully resembling their original versions.

The *YouTube Covers* dataset has 50 original songs from different music genres such as *reggae*, *jazz*, *rock* and *pop music* accompanied of cover versions. In our experiments, we divide this dataset in training and test data. The training data have the original recording in studio and a live version for each music. In the test data, each music has 5 different cover versions that include versions of different music styles, acoustic versions, live performances of established artists, fan videos, etc. Thus, this dataset have a total of 350 songs (100 examples for training and 250 for test). A complete description of *YouTube Covers* dataset is available in our website.

As the *123 Classical* dataset doesn't have a natural division in training/test sets and has a reduced amount of data, we conducted our experimental evaluation in this dataset using stratified random sampling with 1/3 of data to training and the remaining for test. With this procedure, the number of examples per class in the training phase varies from 1 to 5.

5.2 Evaluation Scenarios

In this paper, we consider two different scenarios to evaluate our method: *i) test set as query* and *ii) training set as*

query. In both, the first stage finds shapelets in the training partition.

In the first scenario, we perform a query when a new song arrives. This setting simulates the scenario in which we would like to know if the (test) query is a cover of some previously labeled song. In other words, we use the unlabeled recordings to find similar labeled ones.

In the second scenario, we simulate the scenario in which the author of one of the training songs wants to know if there are uncertified versions of his/her music in the repository. Thus, we should use his/her original recording as query. Therefore, the training instances are used as queries and we use the shapelets to return unlabeled songs that are potentially covers.

5.3 Experimental Setup

In order to evaluate the proposed method, we compare its results against two competitors. The first one is the DTW alignment of the feature vector representing the whole music. The second one uses a music summarization algorithm to find significant segments of the recordings. For this, we use a method that considers that the most significant excerpts of music are those that are most repeated [2]. After finding such excerpts, the similarity search occurs as proposed in this paper.

As feature sets, we used the chroma energy normalized statistics (CENS), as well as chroma extracted together the beat estimating. In general, CENS results are slightly better. Thus, we focus our evaluation using this feature. To extract the CENS, we used the Matlab implementation provided by the Chroma Toolbox [7] with the default parameters settings.

We used the optimal transposition index (OTI) technique to improve robustness for key variances. Shapelets are not used to decide the shift to provide such an invariance. This is done by using the harmonic pitch class profiles (HPCP) of the complete chroma vector.

Our proposal have two parameters related to the windowing: i) window length and ii) overlapping proportion of consecutive windows. For the first parameter, we use the values 25, 50 and 75 for shapelets and 25 for triplets. For the second parameter, we use 2/3 of the window length as overlapping proportion

To provide an intuition to the reader about the first parameter. The mean length of the chroma feature vectors in the datasets *123 Classical* and *YouTube Covers* are 215 and 527, respectively. Therefore, a window length of 25 represents approximately 11% and 5%, respectively, of the average length of the recordings in these datasets.

5.4 Evaluation Measures

In order to assess the quality of our proposal, we used three evaluation measures adopted by MIREX² for the cover song identification task. Such measures take into account the position of the relevant songs in the estimated ranking of similarity.

² http://www.music-ir.org/mirex/wiki/2015:Audio_Cover_Song_Identification

Given a set of n query songs, a retrieval method returns a rank r_i ($i = 1, 2, \dots, n$) for each of them. The function $\Omega(r_{i,j})$ returns the value 1 if the j -th-ranked song obtained for the i -th query is a relevant song or 0 otherwise. In the context of this work, a relevant song is a cover version of the query recording.

The first evaluation measure represents the mean number of relevant songs retrieved among the top ten positions of the ranking (MNTop10). Formally, the MNTop10 is defined according to Equation 2.

$$MNTop10 = \frac{1}{n} \sum_{i=1}^n \sum_{j=1}^{10} \Omega(r_{i,j}) \quad (2)$$

The mean average precision (MAP) is the mean value of the average precision (AP) for each query song. The AP is defined in Equation 3.

$$AP(r_i) = \frac{1}{n} \sum_{j=1}^n \left[\Omega(r_{i,j}) \left(\frac{1}{j} \sum_{k=1}^j \Omega(r_{i,k}) \right) \right] \quad (3)$$

Finally, we also use the mean rank of first correctly identified cover (MFRank). In other words, this measure estimates, on average, the number of songs we need to examine in order to find a relevant one. The MFRank is defined by Equation 4.

$$MFRank = \frac{1}{n} \sum_{i=1}^n fp(r_i) \quad (4)$$

where $fp(r_i)$ is a function that returns the first occurrence of a relevant object in the ranking r_i .

For the first two measures, larger values represent better performance. For the last one the smaller values are indicative of superiority.

5.5 Results

In the Section 4, we proposed several adaptations to the original shapelets approach to the music retrieval setting. Unfortunately, due to lack of space, we are not able to show detailed results for all combinations of these techniques. In total, we have 16 different combinations of techniques. All those results are available on the website created for this work [15].

In this section, we present a subset of the results according to the following criteria:

- **OTI.** We show all results with OTI as key invariance method. For the dataset *YouTube Covers*, the use of OTI led to significant improvements. For the *123 Classical* dataset, OTI performed quite similarly to the same method without OTI. This may occur because the problem of key variations is more evident in the pop music. We notice we used the simplest version of OTI, that assesses just one tonal shift.
- **Shapelet evaluation.** We evaluate all results with DistDiff. In most cases, information gain performed

worst than DistDiff. Even more, there are cases where the use of IG causes a significant performance deterioration. For example, when using a single shapelet per recording on *YouTube Covers*, the method using information gain achieved MNTop10 = 0.75, MAP = 25.29% and MFRank = 17.52. By changing this measure by the DistDiff criterion, proposed in this paper, the results become MNTop10 = 1.22, MAP = 47.14% and MFRank = 9.72.

- **Triplet.** We show the results using triplets. In general the use of a single shapelet to describe the training songs did not outperform the use of triplets. Although obtain an improvement in isolated cases, the differences are small in these cases.

Therefore, we will fix our analysis to the methods that use OTI and triplets evaluated by DistDiff criterion. The last remaining decision concerns the use of Euclidean or DTW distances. We show the results obtained with both.

Table 1 shows the results obtained on *123 Classical* dataset and Table 2 shows the results obtained on *YouTube Covers* dataset.

Table 1. Results achieved on the dataset *123 Classical*

Scenario 1 - Test set as query			
	MNTop10	MAP (%)	MFRank
DTW	2.34	97.24	1.12
Summarization	2.27	93.46	1.00
Triplets-DTW	2.39	97.24	1.02
Triplets-ED	2.38	98.05	1.00
Scenario 2 - Training set as query			
	MNTop10	MAP (%)	MFRank
DTW	4.73	98.92	1.00
Summarization	4.44	91.52	1.02
Triplets-DTW	4.78	99.41	1.00
Triplets-ED	4.71	97.92	1.00

Table 2. Results achieved on the dataset *YouTube Covers*

Scenario 1 - Test set as query			
	MNTop10	MAP (%)	MFRank
DTW	1.14	42.49	11.69
Summarization	0.85	32.11	13.82
Triplets-DTW	1.29	45.55	8.45
Triplets-ED	1.26	47.80	8.49
Scenario 2 - Training set as query			
	MNTop10	MAP (%)	MFRank
DTW	2.11	39.19	6.58
Summarization	1.66	29.20	14.46
Triplets-DTW	2.82	52.87	4.65
Triplets-ED	2.87	54.95	5.18

5.6 Discussion

The results show that triplets outperformed similarity estimation by using music summarization and achieved equal or better results than the DTW matching of the whole feature vector.

More importantly, we notice that the querying using shapelets is significantly more efficient than the matching between the whole songs. Although our method requires a training phase that is absent in similarity search with DTW, such a phase is performed only once.

Let l and m be the length of feature vectors of the query and the labeled songs. The complexity to find an alignment based on dynamic programming, such as DTW, is $\mathcal{O}(lm)$. Now, let s be the size of each shapelet of the training song. The complexity to calculate the shapelet-based Euclidean distance between the query and the original song is $\mathcal{O}(ls)$, with $s \ll m$.

Table 3 shows the time in seconds to perform the retrieval step using Triplets-ED and DTW matching the entire feature vectors.

Table 3. Total time (in seconds) to calculate the distance between all the queries (test set) and the training set by using DTW and Triplets-ED

	Dataset	
	<i>123 Classical</i>	<i>YouTube Covers</i>
DTW	2,294	14,124
Triplets-ED	148	928

The result of this experiment shows that our method is about 15 times faster to retrieve music by similarity. We argue that our method may be further faster with the use of techniques to speed-up the similarity search – to find the best match between the shapelet and the whole feature vector.

The identification rates were similar for both triplets approaches, alternating the best results between them. Although the time spent to calculate Triplets-DTW is potentially lower than the obtained by a straightforward implementation of Euclidean distance [10], the time spent by our simple implementation is similar to the DTW alignment of the whole feature vector.

6. CONCLUSION

In this paper, we propose a novel technique to content-based music retrieval. Our method is naturally invariant to structure and open to aggregate invariance to key and tempo by the choice of appropriate methods, such as OTI and CENS as feature vector.

We evaluated our method in a cover song recognition scenario. We achieved better results than the widely applied approach of DTW alignment and a similar approach based on a well-known summarization algorithm. Our method is also more than one order of magnitude faster than these methods.

There are several possible extensions for this work. For instance, we can extend our idea to a shapelet-transform [6]. The evaluated scenario also suggests research on incremental learning of shapelets, the retrieval considering that novel songs may arrive, among other tasks. Finally, we intend to investigate how to improve the time cost of DTW similarity search in order to make the time of Triplets-DTW be competitive with Triplets-ED.

7. ACKNOWLEDGMENTS

The authors would like to thank FAPESP by the grants #2011/17698-5, #2013/26151-5, and 2015/07628-0 and CNPq by grants 446330/2014-0 and 303083/2013-1.

8. REFERENCES

- [1] Juan Pablo Bello. Measuring structural similarity in music. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2013–2025, 2011.
- [2] Matthew L. Cooper and Jonathan Foote. Automatic music summarization via similarity analysis. In *International Society for Music Information Retrieval Conference*, 2002.
- [3] Emanuele Di Buccio, Nicola Montecchio, and Nicola Orio. A scalable cover identification engine. In *International Conference on Multimedia*, pages 1143–1146, 2010.
- [4] Daniel P. W. Ellis and Graham E. Poliner. Identifying cover songs with chroma features and dynamic programming beat tracking. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages 1429–1432, 2007.
- [5] Emilia Gómez and Perfecto Herrera. Estimating the tonality of polyphonic audio files: Cognitive versus machine learning modelling strategies. In *International Society for Music Information Retrieval Conference*, pages 92–95, 2004.
- [6] Jason Lines, Luke M. Davis, Jon Hills, and Anthony Bagnall. A shapelet transform for time series classification. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 289–297, 2012.
- [7] Meinard Müller and Sebastian Ewert. Chroma Toolbox: MATLAB implementations for extracting variants of chroma-based audio features. In *International Society for Music Information Retrieval Conference*, pages 1–6.
- [8] Meinard Müller, Frank Kurth, and Michael Clausen. Audio matching via chroma-based statistical features. In *International Society for Music Information Retrieval Conference*, pages 288–295, 2005.
- [9] Bee Suan Ong. *Structural Analysis and Segmentation of Music Signals*. PhD thesis, 2007.
- [10] Thanawin Rakthanmanon, Bilson Campana, Abdullah Mueen, Gustavo Batista, Brandon Westover, Qiang Zhu, Jesin Zakaria, and Eamonn Keogh. Searching and mining trillions of time series subsequences under dynamic time warping. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 262–270, 2012.
- [11] Joan Serra, Emilia Gómez, and Perfecto Herrera. Transposing chroma representations to a common key. In *IEEE CS Conference on The Use of Symbols to Represent Music and Multimedia Objects*, pages 45–48, 2008.
- [12] Joan Serra, Emilia Gómez, and Perfecto Herrera. Audio cover song identification and similarity: background, approaches, evaluation, and beyond. In *Advances in Music Information Retrieval*, pages 307–332. Springer, 2010.
- [13] Joan Serra, Xavier Serra, and Ralph G. Andrzejak. Cross recurrence quantification for cover song identification. *New Journal of Physics*, 11(9):093017, 2009.
- [14] Carlos Nascimento Silla Jr, Alessandro Lameiras Koricich, and Celso A. A. Kaestner. The latin music database. In *International Society for Music Information Retrieval Conference*, pages 451–456, 2008.
- [15] Diego F. Silva, Vinícius M. A. Souza, and Gustavo E. A. P. A. Batista. Website for this work – <https://sites.google.com/site/ismir2015shapelets/>.
- [16] Lexiang Ye and Eamonn Keogh. Time series shapelets: a new primitive for data mining. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 947–956, 2009.

IMPROVING SCORE-INFORMED SOURCE SEPARATION FOR CLASSICAL MUSIC THROUGH NOTE REFINEMENT

Marius Miron Julio José Carabias-Orti Jordi Janer

Music Technology Group, Universitat Pompeu Fabra

maris.miron, julio.carabias, jordi.janer@upf.edu

ABSTRACT

Signal decomposition methods such as Non-negative Matrix Factorization (NMF) demonstrated to be a suitable approach for music signal processing applications, including sound source separation. To better control this decomposition, NMF has been extended using prior knowledge and parametric models. In fact, using score information considerably improved separation results. Nevertheless, one of the main problems of using score information is the misalignment between the score and the actual performance. A potential solution to this problem is the use of audio to score alignment systems. However, most of them rely on a tolerance window that clearly affects the separation results. To overcome this problem, we propose a novel method to refine the aligned score at note level by detecting both, onset and offset for each note present in the score. Note refinement is achieved by detecting shapes and contours in the estimated instrument-wise time activation (gains) matrix. Decomposition is performed in a supervised way, using training instrument models and coarsely-aligned score information. The detected contours define time-frequency note boundaries, and they increase the sparsity. Finally, we have evaluated our method for informed source separation using a dataset of Bach chorales obtaining satisfactory results, especially in terms of SIR.

1. INTRODUCTION

Sound source separation has been actively addressed during the recent years with various applications ranging from predominant melody transcription [10], to interference removal in close microphone recordings [4]. State of the art systems particularly target the separation of the predominant harmonic instrument from the accompaniment [3, 4, 10, 18], and less often the separation of various instruments in classical music [9, 15].

Besides [18](recurrent neural networks), and [9](particle filters), the aforementioned systems are based on non-negative matrix factorization (NMF) [19], a technique that efficiently decomposes a magnitude spectrogram into a set


of template (basis) and activation (gains) vectors. However, when dealing with a non-convex problem, the NMF can converge to a local minima solution for which the sources are not well separated. Towards a better separation, the system can benefit from prior knowledge. On this account, a set of musical meaningful variables are introduced into the parametric model and estimated jointly.

Furthermore, important improvements are reported when score information is added to guide the decomposition process [3, 9, 12, 15, 17]. In this case, the best performance is achieved when the audio is perfectly aligned with the score [23]. However, in a real-case scenario, a perfect aligned score is not available, and a score-alignment algorithm is needed [5, 8, 9, 13].

Conversely, as enounced in [3], besides the global misalignments, fixed by score-alignment systems, we can also encounter local misalignments. With respect to this problem, source separation systems propose to estimate the onset implicitly into the parametric NMF model, by increasing the time boundaries for the onsets in the gains matrix at the initialization stage [12, 15, 17]. However, an interesting question is whether such an initialization results in a better separation than refining the gains and correcting the local misalignments prior to the source separation.

Several methods deal with explicitly correcting local misalignments [21, p. 103], [20,27]. The latter finds shapes and contours (blobs) in a pitch salience function, obtained by pre-processing the spectrogram of the signal and then filtering the spectral peaks for each instrument. However, this method does not use any information regarding the timbre, which is more desirable when distributing energy between different instruments.

The goal of this paper is to use the note refinement information in order to improve score-informed source separation of harmonic instruments. Specifically, we have two contributions: we adapt the source separation framework in [24] to the score-informed case, and, notably, we correct the local misalignments in the score and refine the time-frequency zones of the gains used in source separation. First, we compute the initial gains by distributing the energy among instruments with the source separation NMF algorithm proposed in [24]. The computed gains offer a more robust representation than the pitch salience used in [20], because timbre information is used to deal with the problem of overlapping partials between the instruments, and because the gains are represented on log-frequency scale and are less noisy than the pitch salience

 © Marius Miron, Julio José Carabias-Orti, Jordi Janer. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Marius Miron, Julio José Carabias-Orti, Jordi Janer. "Improving score-informed source separation for classical music through note refinement", 16th International Society for Music Information Retrieval Conference, 2015.

in [20]. As a result, detecting and assigning blobs to notes in the gains matrix can be done more robustly. Second, we can use the processed gains to reiterate the NMF source separation. Consequently, instead of initializing the NMF with the MIDI information, we can use the blobs associated with each note. On this account, we restrict the potential interferences not only in time but also in frequency, and achieve better separation.

We evaluate the note refinement and the source separation on the Bach10 dataset [9]. Accordingly, note refinement is performed on an artificially generated score with local misalignments, and on the output the DTW based score alignment algorithm [5]. Furthermore, we evaluate the score-informed source separation, as we want more insight on which initialization method yields better source separation.

The remainder of this paper is structured as follows. First, we describe the existing source separation framework and then, in Section 3, the note refinement method and its application to monaural score informed source separation. Then, we evaluate score alignment and source separation. Finally, we discuss the results and restate the contributions to prior work.

2. NMF FOR SOURCE SEPARATION

In this section we explain the Source Separation Framework used for sound source separation. Further information can be found in [24].

2.1 Signal Model

Techniques based on Non-negative Matrix Factorization (NMF) can be used to efficiently decompose an audio spectrogram as a linear combination of spectral basis functions. In such a model, the short-term magnitude (or power) spectrum of the signal $x(f, t)$ in time-frame t and frequency f is modeled as a weighted sum of basis functions as:

$$x(f, t) \approx \hat{x}(f, t) = \sum_{n=1}^N b_n(f) g_n(t), \quad (1)$$

where $g_n(t)$ is the gain of the basis function n at frame t , and $b_n(f)$, $n = 1, \dots, N$ are the bases. Note that model in eq. (1) only holds under the assumption of a) strong sparsity (only one source active per time-frequency (TF) bin) or b) local stationarity (only for power spectrogram) [2].

When dealing with musical instrument sounds, it is natural to assume that each basis function represents a single pitch, and the corresponding gains contain information about the onset and offset times of notes having that pitch [4]. Besides, restricting the model in (1) to be harmonic is particularly useful for the analysis and separation of musical audio signals since each basis can define a single fundamental frequency and instrument. Harmonicity constrained basis functions are defined as:

$$b_{j,n}(f) = \sum_{h=1}^H a_{j,n}(h) G(f - h f_0(n)), \quad (2)$$

where $b_{j,n}(f)$, are the bases for each note n of instrument j , $n = 1, \dots, N$ is defined as the pitch range for instrument $j = 1, \dots, J$, where J is the total number of instruments present in the mixture, $h = 1, \dots, H$ is the number of harmonics, $a_{j,n}(h)$ is the amplitude of harmonic h for note n and instrument j , $f_0(n)$ is the fundamental frequency of note n , $G(f)$ is the magnitude spectrum of the window function, and the spectrum of a harmonic component at frequency $h f_0(n)$ is approximated by $G(f - h f_0(n))$. Therefore, the harmonic constrained model for the magnitude spectra of a music signal is defined as:

$$\hat{x}(f, t) = \sum_{j=1}^J \sum_{n=1}^N \sum_{h=1}^H g_{j,n}(t) a_{j,n}(h) G(f - h f_0(n)), \quad (3)$$

where the time gains $g_{j,n}(t)$ and the harmonic amplitudes $a_{j,n}(h)$ are the parameters to be estimated.

2.2 Augmented NMF for Parameter Estimation

Non-negativity of the parameters is a common restriction imposed to the signal decomposition method for music signal processing applications. Furthermore, the factorization parameters of equation (3) are estimated by minimizing the reconstruction error between the observed $x(f, t)$ and the modeled $\hat{x}(f, t)$ spectrograms, using a cost function, which is this case the Beta-divergence [14]:

$$D_\beta(x|\hat{x}) = \begin{cases} \frac{1}{\beta(\beta-1)} (x^\beta + (\beta-1)\hat{x}^\beta - \beta x \hat{x}^{\beta-1}) & \beta \in (0, 1) \\ x \log \frac{x}{\hat{x}} - x + \hat{x} & \beta = 1 \\ \frac{x}{\hat{x}} + \log \frac{x}{\hat{x}} - 1 & \beta = 0 \end{cases} \cup (1, 2] \quad (4)$$

For particular values of β , Beta-divergence includes in its definition the most popular cost functions, Euclidean (EUC) distance ($\beta = 2$), Kullback-Leibler (KL) divergence ($\beta = 1$) and the Itakura-Saito (IS) divergence ($\beta = 0$). The parameters in (1) are estimated with an iterative cost minimization algorithm based on multiplicative update (MU) rules, as discussed in [19]. Under these rules, $D(x(f, t)|\hat{x}(f, t))$ does not increase with each iteration while ensuring the non-negativity of the bases and the gains. These MU rules are obtained applying diagonal rescaling to the step size of the gradient descent algorithm (see [19] for further details).

Lets denote as θ_l the parameter to be estimated. Then, the MU rule for θ_l is obtained by computing the derivative $\nabla_{\theta_l} D$ of the cost function with respect to θ_l . This derivative can be expressed as a difference between two positive terms $\nabla_{\theta_l}^+ D$ and $\nabla_{\theta_l}^- D$ [25] and thus, the update rule for parameter θ_l can be expressed as:

$$\theta_l \leftarrow \theta_l \frac{\nabla_{\theta_l}^- D(x(f, t)|\hat{x}(f, t))}{\nabla_{\theta_l}^+ D(x(f, t)|\hat{x}(f, t))}. \quad (5)$$

2.3 Timbral Informed Signal Model

As showed in [6], when appropriate training data are available, it is useful to learn the instrument-dependent bases in

advance and keep them fixed during the analysis of the signals. In the commented work, the amplitudes of each note of each musical instrument $a_{j,n}(h)$ are learnt by using the RWC database [16] of solo instruments playing isolated notes together with their ground-truth transcription. Thus, gains are set to unity for each pitch at those time frames where the instrument is active while the rest of the gains are set to zero. Note that gains initialized to zero remain zero because of the multiplicative update rules, and therefore the frame is represented only with the correct pitch.

The MU rules are computed from equation (5) for the amplitude coefficients and the gains as

$$a_{j,n}(h) \leftarrow a_{j,n}(h) \frac{\sum_{f,t} x(f,t) \hat{x}(f,t)^{\beta-2} g_{j,n}(t) G(f-hf_0(n))}{\sum_{f,t} \hat{x}(f,t)^{\beta-1} g_{j,n}(t) G(f-hf_0(n))} \quad (6)$$

$$g_{j,n}(t) \leftarrow g_{j,n}(t) \frac{\sum_{f,m} x(f,t) \hat{x}(f,t)^{\beta-2} a_{j,n}(h) G(f-hf_0(n))}{\sum_{f,m} \hat{x}(f,t)^{\beta-1} a_{j,n}(h) G(f-hf_0(n))} \quad (7)$$

Finally, the training procedure is summarized in Algorithm 1.

Algorithm 1 Instrument modeling algorithm

- 1 Compute $x(f, t)$ from a solo performance for each instrument in the training database
 - 2 Initialize gains $g_{j,n}(t)$ with the ground truth transcription $R_{j,n}(t)$ and $a_{j,n}(h)$ with random positive values.
 - 3 Update the gains using eq. (6).
 - 4 Update the bases using eq. (7).
 - 5 Repeat steps 2-3 until the algorithm converges (or maximum number of iterations is reached).
 - 6 Compute basis functions $b_{j,n}(f)$ for each instrument j using eq. (2).
-

The training algorithm obtains an estimation of the basis functions $b_{j,n}(f)$ required at the factorization stage for each instrument. Since the instrument dependent basis functions $b_{j,n}(f)$ are held fixed, the factorization can be reduced to the estimation of the gains $g_{j,n}(t)$ for each of the trained instruments j .

2.4 Gains estimation

Here, the classical augmented NMF factorization with MU rules is applied to estimate the gains corresponding to each source j in the mixture. The process is detailed in Algorithm 2.

Algorithm 2 Gain Estimation Method

- 1 Initialize $b_{j,n}(f)$ with the values learned in section 2.3. Use random positive values to initialize $g_{j,n}(t)$.
 - 2 Update the gains using eq. (7).
 - 3 Repeat step 2 until the algorithm converges (or maximum number of iterations is reached)
-

2.5 From the estimated gains to the separated signals

In this work, we assume that the individual sources $y_j(t)$, $j = 1 \dots J$ that compose the mixed signal $x(t)$ are

linearly mixed, so $x(t) = \sum_{j=1}^J y_j(t)$. Lets denote the power spectral density of source j at TF bin (f, t) as $|X_j(t, f)|^2$, $j = 1 \dots J$, then, each ideally separated source $y_j(t)$ can be estimated from the mixture $x(t)$ using a generalized time-frequency Wiener filter over the Short-Time Fourier Transform (STFT) domain as in [14, 15].

Here we use the Wiener filter soft-masking strategy as in [24]. In particular, the soft-mask α_j of source j represents the relative energy contribution of each source to the total energy of the mixed signal $x(t)$ and is obtained as:

$$\alpha_j(t, f) = \frac{\hat{Y}_j(t, f)^2}{\sum_j \hat{Y}_j(t, f)^2} \quad (8)$$

where $\hat{Y}_j(t, f)$ is the estimated source magnitude spectrogram computed as $\hat{Y}_j(t, f) = g_{n,j}(t) b_{j,n}(f)$, $g_{n,j}$ are the gains estimated in Section 2.4 and $b_{j,n}(f)$ are the fixed basis functions learnt in Section 2.3.

Then, the magnitude spectrogram $\hat{X}_j(t, f)$ is estimated for each source j as:

$$\hat{X}_j(t, f) = \alpha_j(t, f) \cdot X(t, f) \quad (9)$$

where $X(t, f)$ is the complex-valued STFT of the mixture at TF bin (t, f) .

Finally, the estimated source $\hat{y}_j(t)$ is computed with the inverse overlap-add STFT over $\hat{X}_j(f, t)$, with the phase spectrogram of the original mixture.

3. PROPOSED METHOD

We adapt the source separation framework described in Section 2 to the score-informed scenario. The framework is initialized with the gains $g_{j,n}^{init}(t)$ derived from a MIDI score having alignment errors. Next, the resulting gains after the NMF separation $g_{j,n}(t)$ are refined with a set of image processing heuristics which we describe in the Section 3.2. Finally, the refined gains $p_{j,n}(t)$ are used to reinitialize the framework and reiterate the separation, towards a better result.

3.1 Score-informed gains computation

We use as input a coarsely aligned score and the associated audio recording. The MIDI score has local misalignments up to d frames for the onset and the offset times. Thus, we initialize the source separation system in Section 2 with the MIDI notes by adding d frames before the onset and after the offset. Consequently, for an instrument j , and all the bins in a semitone n associated with a MIDI note (Figure 1B), we set the matrix $g_{j,n}^{init}(t)$ to 1 for the frames where the MIDI note is played as well as for the d frames around the onset and the offset of the MIDI note. The other values in $g_{j,n}^{init}(t)$ are set to 0 do not change during computation, while the values set to 1 evolve according to the energy distributed between the instruments. The final gains are computed with the algorithm described in Section 2.4, obtaining $g_{j,n}(t)$, the gains which will be used during the note refinement stage (Figure 1C).

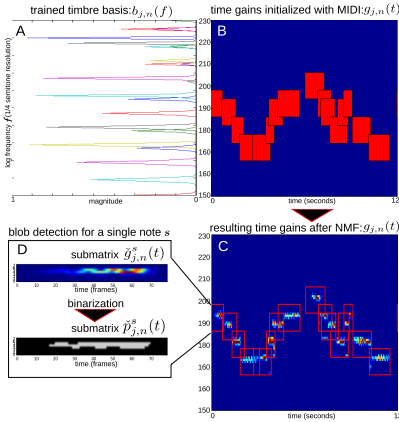


Figure 1. A. The reconstructed signal can be seen as the product between the several harmonic components (A) and the gains (B). After NMF, the resulting gains (C) are split in submatrices and used to detect blobs (D).

3.2 Note refinement

The shape and contours detected in an image, and associated with meaningful objects, are commonly known as blobs [22, p. 248]. Additionally, if we consider the matrix associated with a grayscale image, an image patch is any submatrix of the corresponding matrix.

During the note refinement stage we apply image processing on the gains matrix $g_{j,n}(t)$ in order to associate the entities in an image, namely the blobs, with notes. The chosen blobs give the onset and offset times. Additionally, the areas of the blobs are used to reiterate the separation.

The refinement of the gains occurs for each note separately. Hence, for each note s from the input score we choose an image patch centered at the semitone n corresponding to its associated MIDI note value. Precisely, we process a submatrix of $g_{j,n}(t)$, namely $\tilde{g}_{j,n}^s(t)$, for $s = 1 \dots S$, where S is the total number of notes in the score for an instrument j . The size of submatrix $\tilde{g}_{j,n}^s(t)$, as seen in Figure 1D, is equal to the one of the submatrices which has been set to 1 at the initialization for the corresponding note s . Thus, $\tilde{g}_{j,n}^s(t)$ has a width of two semitones and a length corresponding to the prolonged duration of the note s .

3.2.1 Image binarization

Each image patch is preprocessed in two steps before binarization. Initially, each row vector of the submatrix $\tilde{g}_{j,n}^s(t)$ is convolved with a smoothing gaussian filter to remove noise and discontinuities. Then each column of the same submatrix is multiplied with a gaussian centered at the central frequency bin, in order to penalize the values far from the central bin, but still to preserve vibratos or transitions between notes.

First, we apply a smoothing filter [22, p. 86] on the image patch. We choose a one dimension Gaussian filter:

$$w(t) = \frac{1}{\sqrt{2\pi}\phi} e^{-\frac{t^2}{2\phi^2}} \quad (10)$$

where t is the time axis and $\phi = 3$ is the standard deviation. The first and the last σ elements of each row vector n of the matrix $\tilde{g}_{j,n}^s(t)$ are mirrored at the beginning, respectively at the end of the vector. Then each row vector of $\tilde{g}_{j,n}^s(t)$ is convolved with $w(t)$, and the result is truncated in order to preserve the dimensions of the initial matrix by removing the mirrored frames.

Second, we multiply $\tilde{g}_{j,n}^s(t)$ with a 1-dimensional gaussian centered in the central frequency bin:

$$v(n) = \frac{1}{\sqrt{2\pi}\nu} e^{-\frac{(n-\kappa)^2}{2\nu^2}} \quad (11)$$

where n is the frequency axis, $\kappa = 4$ is the position of the central frequency bin and the standard deviation $\nu = 4$ (one semitone). Then, each column vector of $\tilde{g}_{j,n}^s(t)$ is multiplied with $v(n)$.

Image binarization assumes calculating a submatrix $\tilde{p}_{j,n}^s(t)$, associated with note s :

$$\tilde{p}_{j,n}^s(t) = \begin{cases} 0 & \text{if } \tilde{g}_{j,n}^s(t) < \text{mean}(\tilde{g}_{j,n}^s(t)) \\ 1 & \text{if } \tilde{g}_{j,n}^s(t) \geq \text{mean}(\tilde{g}_{j,n}^s(t)) \end{cases} \quad (12)$$

3.2.2 Blob selection

For a note s we detect blobs the corresponding binary submatrix $\tilde{p}_{j,n}^s(t)$, using the connectivity rules described in [22, p. 248] and [20].

Furthermore, we need to determine the best blob for each note. A simple solution is to compute a score for each blob by summing all the values in $\tilde{g}_{j,n}^s(t)$ included in the area associated with the blob. However, we want to penalize parts of the blobs which overlap in time with other blobs from different notes $s - 1, s, s + 1$. Basically, we want to avoid picking the same blobs for two adjacent notes. Thus, we weight each element in $\tilde{g}_{j,n}^s(t)$ with a factor γ , depending on the amount of overlapping with blobs from adjacent notes, and we build a score matrix:

$$\tilde{q}_{j,n}^s(t) = \begin{cases} \gamma * \tilde{g}_{j,n}^s(t) & \text{if } \tilde{p}_{j,n}^s(t) \wedge \tilde{p}_{j,n}^{s-1}(t) = 1 \\ \gamma * \tilde{g}_{j,n}^s(t) & \text{if } \tilde{p}_{j,n}^s(t) \wedge \tilde{p}_{j,n}^{s+1}(t) = 1 \\ \tilde{g}_{j,n}^s(t) & \text{otherwise} \end{cases} \quad (13)$$

where γ is a value in the interval $0..1$.

Note that we do not use the dynamic programming method in [20] because the images patches are small, thus we have to choose between very few blobs and, to that respect, the Dijkstra algorithm is superfluous.

Furthermore, we compute a score for each note s and for each blob associated with the note, by summing up the elements in the score matrix $\tilde{q}_{j,n}^s(t)$ which are a part of a blob. Furthermore, the selected blob for a note s is the one having the maximum score. The boundaries of the selected blob give the note onset and offset. Additionally, the area of the blob can be used to reiterate source separation.

3.3 Extension to score informed source separation

Our assumption is that better alignment gives a more sparse initialization of the gains $g_{j,n}(t)$, which limits the way energy distributes along instruments during the NMF, and yields better source separation. Additionally, we can further increase sparsity by knowing the frequency boundaries of the notes and by initializing the gains with the detected blob contours. However, by limiting the areas in the activations to the area of the chosen blobs, we discard energy from the unchosen blobs. This energy which is discarded from an instrument can be redistributed between the other instruments by reiterating the factorization.

Let $p_{j,n}^s(t)$ be the matrices derived from the submatrices $\tilde{p}_{j,n}^s(t)$, containing 1 for the elements associated with the selected blob for the note s and 0 otherwise. Then, the new matrix $g_{j,n}(t)$ can be formed with the submatrices $p_{j,n}^s(t)$. For the corresponding bins n and time frames f of a note s , we initialize the values in $g_{j,n}(t)$ with the values in $p_{j,n}^s(t)$. Subsequently, we repeat the factorization using the timbre-informed algorithm described in Section 2.4, this time initializing it with the refined gains. Moreover, we calculate the spectrogram of the separated sources with the method described in Section 2.5.

4. EXPERIMENTAL SETUP

a) Time-Frequency representation: In this paper we use a low-level spectral representation of the audio data which is generated from a windowed FFT of the signal. A Hanning window with the size of 92 ms, and a hop size of 11 ms are used (for synthetic and real-world signals). Here, a logarithmic frequency discretization is adopted. Furthermore, two time-frequency resolutions are used. First, to estimate the instrument models and the panning matrix, a single semitone resolution is proposed. In particular, we implement the time-frequency representation by integrating the STFT bins corresponding to the same semitone. Second, for the separation task, a higher resolution of $1/4$ of semitone is used, which has proven to achieve better separation results [4]. These time-frequency representations are obtained by integrating the short-term Fourier transform (STFT) bins corresponding to the same semitone, or $1/4$ semitone, interval. Note that in the separation stage, the learnt basis functions $b_{j,n}(f)$ are adapted to the $1/4$ semitone resolution by replicating 4 times the basis at each semitone to the 4 samples of the $1/4$ semitone resolution that belong to this semitone.

b) Dataset: We evaluate the note refinement and the source separation on the Bach10 dataset presented in [9] and comprising ten J.S. Bach chorales played by a quartet (violin, clarinet, tenor saxophone and bassoon), each piece having the duration ≈ 30 seconds. The instruments were recorded separately, then mixed to create a monaural audio sampled at 44.1 kHz. Moreover, the Bach10 dataset has certain traits which influence the note refinement and source separation. For instance, the chorales present a homophonic texture which makes it more difficult when performing source separation. Additionally, the results are

directly related to the tempo of the recordings [9]. For this dataset, the tempo is slower than other classical music pieces, there are very few notes below the quarter note level, and we have prolonged notes, known as fermata.

The audio files are accompanied by two MIDI scores: the perfectly aligned ground truth, and a score which has global and local misalignments. Moreover, in order to test the note refinement we use two datasets. The dataset *disA*, proposed in [20], introduces errors for the ground truth onsets and offsets in the interval [100...200] ms. Additionally, we plan to refine the alignment at the note level for the score alignment method described in [5], denoted as dataset *dtwJ*. The method offers solely note onset information, therefore we use the onset of the next note as the note offset for the current note.

c) Evaluation metrics: For score alignment, we evaluate note onsets and offsets in terms of alignment rate, similarly to [7], ranging from 0 to 1, defined as the proportion of correctly aligned notes in the score within a given threshold. For source separation, the evaluation framework and the metrics are described in [26] and [11]. Correspondingly, we use *Source to Distortion Ratio* (SDR), *Source to Interference Ratio* (SIR), and *Source to Artifacts Ratio* (SAR).

d) Parameters tuning: We picked 50 number of iterations for the NMF, and we experimentally determined value for the beta-divergence distortion, $\beta = 1.3$.

5. RESULTS

5.1 Score alignment

We measure the alignment rate of the input score presenting misalignments (B), the alignment method described in Section 3.2 (E), and the one in [20] (D), on the two datasets "disA" and "dtwJ". We vary the threshold within the interval [15...200]. Subsequently, in Figure 2 we present the results for the datasets "disA" and "dtwJ". The errors of the original scores are presented with dotted and straight black lines. For the aligned onsets, aligned rates are drawn with dashed lines and for offsets with straight lines.

We observe that both refinement methods improve the align rate of the scores with local misalignments (black line). For lower threshold, the proposed method (red) improves the method in [20] (blue). Moreover, considering that offsets are more difficult to align, the proposed alignment outperforms the one in [20] when it comes to detecting offsets, within a larger threshold.

5.2 Source separation

We use the evaluation metrics described in Section 4c. We initialize the gains of the separation framework with different note information, as seen in Figure 3. Specifically, we evaluate the perfect initialization with the ground-truth MIDI, Figure 3(A), with the score having local misalignments (*disA*) or the output of a score alignment system *dtwJ*, Figure 3(B), the common practice of NMF gains initialization in state of the art score-informed source separation [12,15,17], Figure 3(C), and the refinement approaches: Figure 3(D,E,F). Note that in D and E we initialize the

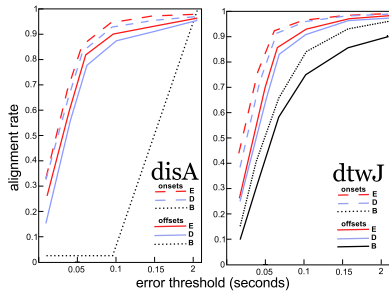


Figure 2. Alignment rate for the two datasets; “B” denotes the score to be refined; “E” and “D” are the scores refined with the methods in Section 3.2 and [20].

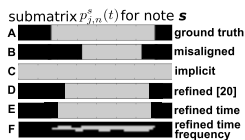


Figure 3. The test cases for initialization of score-informed source separation, for the submatrix $p_{j,n}^s(t)$

gains prior to a note refinement stage with the methods described in [20] (refined [20]) and in the Section 3.2 (refined time), and in F we further refine the gains as proposed in Section 3.3 (refined time frequency).

The results for the test cases A-F, for the two datasets *disA* and *dtwJ* are presented in Table 1 in terms of means of SDR, SIR, SAR. Additionally, audio examples of the separation can be listened online [1].

The proposed system F improves over the other cases in terms of SDR, for all the input scores. Particularly, when we refine the gains in frequency we obtain higher SIR values, hence less interference. Consequently, F yields better results than A, the initialization with ground-truth MIDI annotations, and than E, which is just note refinement in time, without tracking the shape of the blob. On the other hand, the ground-truth A has better SAR values, less artifacts, but has more interference, since F sets to zero some parts of the gains matrix for which the energy does not get redistributed. Additionally, F improves over C, the implicit initialization which extends the time span for the gains, which is the most used approach by the state of the art score-informed source separation algorithms when dealing with local misalignments. On the other hand, the worse decision is not to do any refinement, as in case B.

Moreover, F achieves better results than A-E refining the alignment of [5] (dataset *dtwJ*). However, as this dataset does not have large local misalignments, the difference between F and C, and even B, is not as high as for dataset *disA*, and the improvement is not remarkable. Note that F is better than A in this case as well, suggesting that our

	dataset <i>disA</i>			dataset <i>dtwJ</i>		
	SDR	SIR	SAR	SDR	SIR	SAR
A	6.31	7.10	25.26	6.31	7.10	25.26
B	3.72	4.04	15.20	6.19	6.99	24.59
C	5.18	5.67	19.62	6.25	6.97	25.31
D	5.89	6.80	22.41	5.79	6.67	23.69
E	6.24	7.08	24.43	6.07	6.99	24.58
F	6.35	7.37	24.18	6.37	7.23	25.45

Table 1. Means of SDR, SIR, SAR for the datasets *disA* and *dtwJ* for test cases A-F, for all the instruments

proposed method is robust with regards to different kinds of inputs: significant local misalignments as the dataset *disA*, or smaller as dataset *dtwJ*. Additionally, ground truth offsets are close to the next note onsets, thus *dtwJ* achieves better separation compared to *disA*.

Furthermore, with respect to the performance achieved by other source separation frameworks, tested on the same dataset [9], the results in terms of SDR are similar. The method we propose in this paper is used with the source separation framework [24], but can be adapted to other NMF based frameworks. However, due to the TF representation used in the method, even for ideal TF masks, the separated examples might exhibit cross-talk at high frequencies. This fact is reflected in the measures by lower SIR values.

6. CONCLUSIONS

We proposed a timbre-informed note refinement method to correct local misalignments and to refine the output of state of the art audio-to-score alignment systems, for monaural classical music recordings. We extended the source separation framework proposed in [24] for the case of monaural score informed source separation by refining the gains. The approach increases the sparseness of the gains initialization, achieving better performance than the implicit approach of estimating the onset with a parametric model, as [12, 15, 17], especially for input scores having large local misalignments. Particularly, the proposed system reduces the interference, resulting in higher SIR values. Additionally, the method improves the alignment rate over the one in [20], and is more robust because it uses meaningful timbre information.

As future work, the selection of the best blob and the binarization threshold could be included into the factorization framework through the cost function. Moreover, we plan to test our method with more complex orchestral recordings, and for multi-channel source separation.

7. ACKNOWLEDGEMENTS

This work was supported by the European Commission, FP7 (Seventh Framework Programme), STREP project, ICT-2011.8.2 ICT for access to cultural resources, grant agreement No 601166. Phenix Project

8. REFERENCES

- [1] Bach10 dataset source separation demo. <https://dl.dropboxusercontent.com/u/80928189/demos/index.html>.
- [2] L. Benaroya, F. Bimbot, and R. Gribonval. Audio source separation with a single sensor. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(1):191–199, 2006.
- [3] J.J. Bosch, K. Kondo, R. Marxer, and J. Janer. Score-informed and timbre independent lead instrument separation in real-world scenarios. In *Signal Processing Conference (EUSIPCO)*, pages 2417–2421, Aug 2012.
- [4] J. J. Carabias-Orti, M. Cobos, P. Vera-Candeas, and F. J. Rodríguez-Serrano. Nonnegative signal factorization with learnt instrument models for sound source separation in close-microphone recordings. *EURASIP J. Adv. Sig. Proc.*, 2013:184, 2013.
- [5] J. J. Carabias-Orti, F. J. Rodríguez-Serrano, P. Vera-Candeas, N. Ruiz-Reyes, and F. J. Cañadas-Quesada. An audio to score alignment framework using spectral factorization and dynamic time warping. *ISMIR*, 2015.
- [6] J. J. Carabias-Orti, T. Virtanen, P. Vera-Candeas, N. Ruiz-Reyes, and F. J. Canadas-Quesada. Musical Instrument Sound Multi-Excitation Model for Non-Negative Spectrogram Factorization. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1144–1158, October 2011.
- [7] A. Cont, D. Schwarz, N. Schnell, and C. Raphael. Evaluation of real-time audio-to-score alignment. In *ISMIR*, 2007.
- [8] S. Dixon. Match: A music alignment tool chest. In *ISMIR*, 2005.
- [9] Z. Duan and B. Pardo. Soundprism: An online system for score-informed source separation of music audio. *IEEE Journal of Selected Topics in Signal Processing*, pages 1–12, 2011.
- [10] J.L. Durrieu, A. Ozerov, and C. Févotte. Main instrument separation from stereophonic audio signals using a source/filter model. *EUSIPCO*, (1), 2009.
- [11] V. Emiya, E. Vincent, N. Harlander, and V. Hohmann. Subjective and Objective Quality Assessment of Audio Source Separation. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(7):2046–2057, September 2011.
- [12] S. Ewert and M. Müller. Score-Informed Voice Separation For Piano Recordings. *ISMIR*, pages 245–250, 2011.
- [13] S. Ewert, M. Müller, and P. Grosche. High resolution audio synchronization using chroma onset features. In *ICASSP*, pages 1869–1872. IEEE, 2009.
- [14] C. Févotte, N. Bertin, and J.L. Durrieu. Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis. *Neural computation*, 21(3):793–830, 2009.
- [15] J. Fritsch and M. Plumbley. Score informed audio source separation using constrained non-negative matrix factorization and score synthesis. *ICASSP*, pages 888–891, 2013.
- [16] M. Goto. Development of the rwc music database. In *JCA*, pages 553–556, 2004.
- [17] R. Hennequin, B. David, and R. Badeau. Score informed audio source separation using a parametric model of non-negative spectrogram. *ICASSP*, (1), 2011.
- [18] P.-S. Huang, M. Kim, M. Hasegawa-Johnson, and P. Smaragdīs. Singing-voice separation from monaural recordings using deep recurrent neural networks. In *ISMIR*, 2014.
- [19] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.
- [20] M. Miron, J.J. Carabias, and J. Janer. Audio-to-score alignment at the note level for orchestral recordings. In *ISMIR*, 2014.
- [21] B. Niedermayer. *Accurate Audio-to-Score Alignment Data Acquisition in the Context of Computational Musicology*. PhD thesis, Johannes Kepler Universität, 2012.
- [22] M. Nixon. *Feature Extraction and Image Processing*. Elsevier Science, 2002.
- [23] C. Raphael. A classifier-based approach to score-guided source separation of musical audio. *Comput. Music J.*, 32(1):51–59, March 2008.
- [24] F. J. Rodríguez-Serrano, J. J. Carabias-Orti, P. Vera-Candeas, T. Virtanen, and N. Ruiz-Reyes. Multiple instrument mixtures source separation evaluation using instrument-dependent NMF models. In *LVA/JCA*, pages 380–387, 2012.
- [25] D.L. Sun and C. Févotte. Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence. In *Acoustics, Speech and Signal Processing (ICASSP)*, 2014 *IEEE International Conference on*, pages 6201–6205, May 2014.
- [26] E. Vincent, R. Gribonval, and C. Févotte. Performance measurement in blind audio source separation. *Audio, Speech, and Language Processing, IEEE Transactions on*, 14(4):1462–1469, July 2006.
- [27] S. Wang, S. Ewert, and S. Dixon. Compensating for asynchronies between musical voices in score-performance alignment. In *ICASSP*, pages 589–593, Brisbane, Australia, 2015.

IN THEIR OWN WORDS: USING TEXT ANALYSIS TO IDENTIFY MUSICOLOGISTS' ATTITUDES TOWARDS TECHNOLOGY

Charles Inskip

Department of Information Studies,
University College London
c.inskip@ucl.ac.uk

Frans Wiering

Department of Information and Computing Sciences,
Universiteit Utrecht
f.wiering@uu.nl

ABSTRACT

A widely distributed online survey gathered quantitative and qualitative data relating to the use of technology in the research practices of musicologists. This survey builds on existing work in the digital humanities and provides insights into the specific nature of musicology in relation to use and perceptions of technology. Analysis of the data (n=621) notes the preferences in resource format and the digital skills of the survey participants. The themes of comments on rewards, benefits, frustrations, risks, and limitations are explored using an *h*-point approach derived from applied linguistics. It is suggested that the research practices of musicologists reflect wider existing research into the digital humanities, and that efforts should be made into supporting development of their digital skills and providing usable, useful and reliable software created with a 'musicology-centred' design approach. This software should support online access to high quality digital resources (image, text, sound) which are comprehensive and discoverable, and can be shared, reused and manipulated at a micro- and macro level.

1. INTRODUCTION

In the last two decades, an astonishing amount of computer technologies have been created for the processing of digitized music and music metadata. Quite a few of these are targeted at musicological research. Very often, such software, standards, services or resources are the outcome of interdisciplinary collaborations between computer scientists, audio engineers, musicologists and/or library scientists. An ever-present subtext in the discourse around these collaborations is the potential of technology to transform the discipline of musicology. Yet the uptake of these technologies in mainstream musicology is not widespread. As a first step in a timely systematic exploration of the area, this paper presents the results of a questionnaire amongst musicologists worldwide, focussing on the use or non-use of technology resources in their daily work processes. Gathering insights into the aims and values of the researchers is an important step towards creating a 'musicology-centred' design practice that is founded on human-centred design methods [1]. The key characteristic of such methods is to focus on human work practices and bottlenecks, and then to select or develop the technologies that remove these bottlenecks while respecting the aims and values of the humans in-

involved. Whereas human-centred approaches to systems design are increasingly used in digital humanities, they have been rarely applied to digital musicology.

The use of modern technology in the digital humanities has been widely explored in the last ten years [2-9]. Existing research has identified domain-specific differences between humanities and scientific researchers in their information behaviours. These appear to be predominantly influenced by the analogue or digitised surrogate nature of the research objects in humanities, and the practices of humanities researchers, which are frequently around lone research. Research indicates that humanists welcome technology when it speeds up workflow [8-9], rely on informal peer networks, primarily access monographs, libraries and private collections, search by browsing and citation chasing, and use exploratory search strategies [2]. The core issue underlying technology adoption is thus not so much technophobia as the acceptability and relevance of technology as part of the research process.

This work sets out to explore the adoption of software tools by musicologists in their digital scholarship practices ("the ability to participate in emerging academic, professional and research practices that depend on digital systems" [10]). These tools, which allow the interrogation of digital musical artefacts (including music notation, digital audio, or contextual texts such as metadata) have been widely reported on and refined through the annual ISMIR conference. However it appears that there is some disconnect between this research strand and musicological users' needs and requirements [11-14]. Although some efforts are made to consider user information needs and behaviours [15-19], these are outweighed by a systems-centred approach to the development of new tools [19]. This may reflect the findings that developers determine the success of their efforts more by the performance of the tool than its uptake by users [5, 6]. However, in the words of Borgman [20]: "*until analytical tools and services are more sophisticated, robust, transparent, and easy to use for the motivated humanities researcher, it will be difficult to attract a broad base of interest within the humanities community.*"

Although, for example, the AHRC-funded Transforming Musicology project [21] attempts to encourage closer collaboration between musicologists, computer scientists and software developers, only a few MIR projects seem to be based on an understanding of the work processes and related technology needs of musicologists [22-24]. Building on recent studies into the adoption of tools and resources by humanists [3, 4, 25], this research presents a large-scale investigation of the digital scholarship practices of musicologists. The results will hopefully contrib-



© Charles Inskip, Frans Wiering.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Charles Inskip & Frans Wiering "In their own words: using text analysis to identify musicologists' attitudes towards technology". 16th International Society for Music Information Retrieval Conference, 2015.

ute to the development of usable systems which reflect work practices and attitudes of this community.

2. METHODOLOGY

We created an online survey named ‘What Do Musicologists Do All Day’ (WDMDAD). With this survey we wanted to gather data on the research musicologists do, how they use (or don’t use) technology in their research, and how they assess positive and negative aspects of technology. Our main purpose was to collect rich and detailed stories in their responses, which we did by means of open-ended questions, contextualized within demographic data. We were seeking to explore behaviours and attitudes by encouraging the participants to communicate their experiences more freely than in a multiple choice survey. Our emphasis on rewards, frustrations, risks, limitations and benefits was drawn from a desire to encourage constructive responses of both a positive and negative nature, and enable us to build on previous work in digital humanities, particularly [4]. Though the questions are in English, we encouraged the participants to use their own language if they felt more comfortable this way. The questions are shown in Table 1. The survey rubric and questions were carefully designed to encourage musicologists with a broad range of digital skills and experience to contribute to the survey. Responses are anonymous. All participants gave informed consent in the use of the data they provided, following ethical guidelines of the researchers’ institutions.

The survey was published online using the Opinio system. After the final question, participants were linked to a Google Form, where they were given the option to leave contact details if they wished to be informed about the results or participate in follow-up research. These personal data were not linked to the survey responses, maintaining the researchers’ commitment to anonymity of the participants. The link to the survey was posted on various musicological mailing lists (including AMS, IAML (c. 700 subscribers), ICTM, SMT, musicology-all and several national lists). To stimulate wide international participation a mailing was sent to all (c. 700) members of the International Musicological Society and the Society for Interdisciplinary Music Studies (c. 70 members). Invitations to participate were circulated by national societies or lists in Australia, Austria, France, Germany, Netherlands (c. 200 members) and other countries. WDMDAD was mentioned a few times on social media. It is not known whether all participants are ‘genuinely’ musicologists, but from reviewing the responses it is clear that they self-identified as such. It is also possible that participation was skewed once the survey link was released ‘into the wild’. Responses were collected from 4 December 2014 until 6 March 2015. Initially, there were some technical issues in showing the link to the Google Form, mainly for IOS devices, resolved after a few days. As a consequence, some participants submitted responses multiple times. Duplicate responses were removed, as were responses that didn’t get beyond the first page (Q1-4). There was only one fake response. Responses in languages other than English were translated by native speakers in

collaboration with the research team who were able to provide explanatory context. The cleaned dataset responses were analysed identifying themes and patterns in the data, using a combination of Excel, SPSS and NVivo10.

Question	Response
Q1: What is your gender?	male / female / prefer not to say
Q2: What is your age?	choose one of 6 categories
Q3: Please identify your location from this list	pick country from list
Q4: What is your level of education?	bachelor / masters / PhD / other (specify)
Q5: How confident would you say you are using digital systems and materials to find, organise and analyse research materials, and create and disseminate your findings?	5-point Likert scale (low-high)
Q6: Where do you do your musicology research? (you can choose more than one, if you like)	select from 4 categories, if ‘other’, specify
Q7: What is your speciality? (you can choose more than one, if you like)	select from 11 categories, if ‘other’, specify
Q8: What are you currently researching?	Text
Q9: Which is the information or music resource you use most in your musicology research and writing?	choose one of 10 categories, if ‘other’, specify
Q10: Which [Q9] do you use, why?	text
Q11: If you think you may have a preference for using digital or physical resources in your work, why do you think this is?	text
Q12: Tell a story about a rewarding or a frustrating experience (or both, if you like) with technology in your music research.	text
Q13: What do you think are the risks and limitations of the use of technology in musicology research?	text
Q14: What do you think are the benefits of using technology in musicology research?	text

Table 1. Survey questions

The full texts were imported into NVivo10 for analysis. After automated removal of stop-words, the remaining terms were ranked by frequency. Recognising the importance of frequency in terms of identifying vocabularies and enabling comparisons between texts, recent work in applied linguistics [26] has found some value in applying the Hirsch index (*h*-index) [27] citation measure approach to text analysis. The percentage of appearance of key terms is generally around the 1-2% level, which is not unusual in this type of work. Most words only appear once. The *h*-point (where term rank = term frequency) provides a threshold whereby important thematic words (autosemantics) lying above this point are considered to be more significant than those below the *h*-point. Here, as stop words (synsemantics) had previously been removed from the texts, this approach enabled the identification of high-ranking autosemantics which were more likely to be related to the theme of the text [26] and was preferable to

arbitrarily choosing the ‘top 10/15/20’ terms as it also enabled comparison between texts. Visualisations of the concordances of the terms in the pre-*h* domain were examined to provide insights into their context. This process was repeated for each autosemantic term in the pre-*h* domain for each text (rewards, benefits, risks, limitations, frustrations). There were between 7,300 and 13,000 words in each of these corpora, each containing between 1,500 and 2,400 unique terms.

3. FINDINGS

3.1 Demographics

The data presented here focus on those aspects that are relevant to the analysis presented in this paper. The total number of usable responses was 621, coming from 46 different countries. A large majority of survey participants were from two continents: Europe (306) and North America (248). Around two thirds of the respondents (385) were from English-speaking countries.

Responses span all career phases, with the highest representation of the 30-39 age group (Figure 1). Females (314) and males (294) participated in almost equal numbers (13 prefer not to say). The respondents’ level of education is high, with ‘PhD/Doctorate’ (449) and ‘Masters’ (129) as the largest categories. The two most important locations for doing research are ‘Academic institution’ (493) and ‘Library, archive or museum’ (197).

Digital skills	Count
1	2
2	18
3	132
4	256
5	213

Table 2. Self-evaluated level of digital skills (1=low, 5=high; mean=4.06, n=621)

Respondents assess their digital skills quite highly (Table 2) but there are considerable age differences (Figure 1). Although anecdotally there is a tendency for digital skills to decrease with age, more than half of the 70+ respondents rate their digital skills (DS) at 4 or 5.

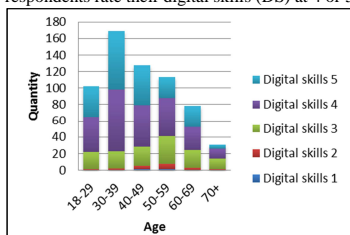


Figure 1. Age group and digital skills of participants (n=621)

3.2 Preferred type of resource

Respondents were asked to choose one type of preferred resource from a list (Table 3). Although some were reluc-

tant to make a choice, overall 319 respondents prefer digital resources, 271 prefer physical resources. Musical resources, whether audio or notation, are preferred by only 43 respondents. However, the responses to Q10 show that a considerable part of the archival and manuscript collections are actually researched for their musical content.

Resource	Count
Digital books and journals	193
Physical books and journals	188
Digitised archives and manuscript collections	104
Physical archives and manuscript collections	62
Other resource	31
Music audio on computer, phone, mobile device	15
Music audio on tape, record, CD	12
Physical collection of music editions	9
Digital collection of music editions	4
Online music audio collection	3

Table 3. Preferred resource (n=621)

It can be seen in Figure 2 that there appears to be a correlation between the preferred format and the level of digital skills, participants with digital skills 3 (DS3) preferring physical resources, while those with 5 (DS5) lean towards digital resources.

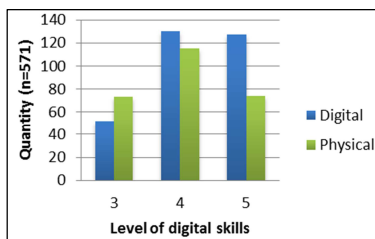


Figure 2. Preferred information resource by digital skills (n=571)

The participants were given the option to choose more than one speciality subject. The majority selected historical musicology. The representations in Figure 3 provide some insights into the self-evaluated digital skills across speciality. While computational and systematic musicology shows a higher coverage of DS4 and DS5, performance practice, historical and library / archive / museum research and other areas of study show a higher proportion of DS1-3.

3.3 Rewards

For Q12, an *h*-point of 23 was identified. Terms from the pre-*h* domain are emboldened hereafter. (Respondent code in parentheses.) **Access**, here, is used in relation not only to access to the researchers’ own materials “almost wherever I am” (091) but more widely to **digitized** primary and secondary **sources** such as “databases, online journals, digitised books, scores” (168), “newspaper archives” (201), “quality recordings” (221) and “high-quality color images” (557). This access allows engagement of a high quality: “It really makes me feel I could be

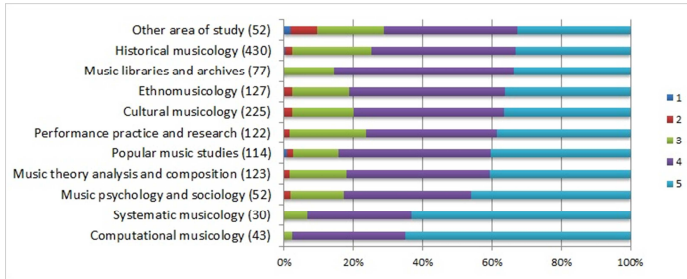


Figure 3. Percentage digital skills per speciality (n=1395) in a library in Italy” (270) and it is not unusual to find this being evaluated favourably in terms of **time**-saved. Further deep analysis through close reading of the texts of the concordances around these key autosemantics highlighted the importance to the participants of using technology to save **time** and increase the speed of their workflow: “...now I can see them all in one afternoon” (022) and minimize the need for travel to engage with a wide range of primary and secondary **sources**. Images of **manuscripts**, scores and digital **books** are considered to be particularly useful, while favourable mentions of **library** catalogues, **digital** archives, scholarly databases and various types of **software** (Sonic Visualiser, Audacity, image manipulation) also feature widely in these texts: “I cannot think what I would be able to do without this software!” (592).

3.3 Benefits

The process was repeated, examining the texts describing the participants views on the benefits of technology (Q14). The pre-*h* domain (*h*=35) vocabulary featured some similar terms to those in the ‘rewards’ texts, but included a richer, less concentrated use of terms, reflected by the higher *h* value. This indicates there is a wider range of issues than in the ‘rewards’ texts. Once again, **access** was considered to be an important term in the vocabulary. It creates the “potential to formulate projects or research questions hitherto unthinkable” (003), saving **time** and money, reducing the need for travel to visit **archives** and improving efficiency, enabling researchers to engage with up-to-date **resources** or **materials** (in the physical form as manuscripts or other paper-based documents, or as **recordings**) located globally which would otherwise be out of reach because of distance, cost, or the fragility of unique items. Downsides are recognized: “it can be really time consuming to separate the wheat from the chaff” (313) and “excess of information, lack of a methodology for analyzing recorded sound” (203). It is not only materials that are accessible: “... people, music, documents, can be accessed around the world” (336). This accessibility enables the collection and analysis of **data** “in a way which would not be possible for a human being” (021) which may lead to “...more accurate findings, as many things can be really ‘counted’, not the gut feeling that musicologists in the past had” (039). Gather-

ing, organising, processing, manipulating and analyzing data are key benefits for some members of this community: “modern technology provides new research opportunities, it helps to work in a time-saving way and it makes communication easier and faster.” (286). The ability to **share** research data, ideas and findings

more easily is also highlighted (“whether it be in formal ‘journal’ form or informal such as facebook, email, or texting” (249).

3.4 Risks

For the texts relating to ‘risks’ the *h*-point was 20. The recurring theme of **access** here (Q13) focuses on how “the vast majority of resources have not been digitized” (65) and the risk of loss of knowledge (through lack of comprehensive digitized collections, or closed subscriptions) and loss of artefacts (through failure of or developments in technology). It is suggested that “immediacy of access to a wide range of material encourages a rapidity of response and decision” (015) which may lead to more superficial research and there are fears that physical objects may even “be overlooked” (319) leading to “privileging digital sources” (188). Some of the views on **access** link to those on **availability**. Excessive amounts of available resources may lead to “complacency and overconfidence” (052), “an incomplete and imbalanced picture” (186) or “laziness” (numerous). There is evidence of strong feelings in these texts that the wide availability of digitized resources may mean that “musicology will be too superficial and lose authority as a serious contribution to society” (604) and that by focusing on electronic journals rather than **books** this may lead to “apparently clever new historicist readings that are in fact shallow.” (424). This links to a strong view that technological determinism is a problem: “Technology ... cannot replace using the grey stuff between the ears” (003). While concerns about the risks of losing or corrupting insufficiently preserved or stored **data** appear, there is a fear that the problem in concentrating on the interpretation of large datasets may be “that is not feasible to listen through and analyze. It disincentivizes selective recording” (312) and “need[s] to be done with extreme care” (100). The tension between **digital** materials and the **materiality** of **physical sources** and **resources** reinforces this apparent fear of superficiality and, particularly, incompleteness of research “[s]ome things cannot be gleaned from digitized copies only” (090). For some, digital materials are not to be trusted because of the “seduction” and “temptation” of their (inherent) “shallowness”. This is not the only view: “From my informatics-biased standpoint, the use of digital technology in music research is a clear net-positive as a way to augment and enhance traditional musicological approaches” (410).

3.5 Limitations

The 'limitations' texts *h*-point was 22. The fears around **materiality** are echoed in the comments on limitations (Q13), partly because increased **access** to the **digital** manifestation of **information** objects can be seen to lead to decreased availability of the physical item, and those which have not yet been **digitized** are also considered to be unavailable. Costly subscriptions to academic journals (JSTOR is particularly popular) are a concern to unaffiliated researchers and those within academia alike (as subscriptions may be limited to on-site access): "*digitization thus increasingly creates a dichotomy of researchers*" (337) and Open Access is not seen to successfully solve this issue. The requirement to have access to the Internet and competency in the use of technology is also seen as a limitation by some. The use of **archives** continues to reflect the concerns around the **materiality** argument and develops on the theme of comprehensive research practices: "*Carl Ludwig's 'Repertorium Organorum' may be hellish to use, but it's still indispensable*" (058). The opportunities for "*serendipity*" through browsing the physical **library** are particularly highlighted: "*Browsing in the digital realm is a far less productive activity than browsing in library stacks*" (068) and digital archives "*do not always capture the creative process, or iterations, of materials*" (420). **Search for sources** may be incomplete, "*missing the surrounding context*" (037) and particularly OCR is limited. It seems likely there is a role for **librarians** here in terms of developing the search skills of their users alongside the functionality of their search interfaces: "*I'm never certain that all bases have been covered in a search*" (233). When speaking about primary research **sources**, again the materiality is paramount: "*It is much easier to turn a page physically*" (341) as well as authority: "*Digital materials can be posted by anyone*" (492). Although the participants generally seem happy to either **read books** online or from the shelf (with some strong exceptions relating to materiality, eye-strain and the tendency to skim electronic materials), they are wary of the problems around e-books' usability and long term access.

3.6 Frustration

Notable in the 'frustrations' texts (Q12) (*h*-point=26) was the appearance of **software** brands, particularly **Final**, **Sibelius**, **Office** and **Word**. These frustrations are important issues when considering the self-assessed digital skills of the participants. Despite most participants describing themselves as being 3 – 5 in digital skills, they are suffering from **software** (or **programmes**) being difficult to integrate with the idiosyncrasies of musical research practices as well as being time-consuming to learn, unreliable and unnecessarily updated. Although users may be familiar with Linux, LaTeX and Sonic Visualiser, some participants are not working with modern **software**.

More generally, **documents** here are generated by the researchers and may be unexpectedly reformatted in some way by software, while **data** can be '*the bane of my existence*' (198) in terms of entering, and is easily lost or corrupted if it has not been backed up (an '*annoyance*' (368)). **Books** (electronic or physical) and **recordings** can be difficult to **find** because **library** catalogues are not

always intuitive, and e-books are difficult to read because library e-reader interfaces are '*unfriendly*' (081) and '*difficult to use*' (086). Hardware can create difficulties – **computers** can be '*very old*', '*slow*', and can '*crash*' – intervention by intermediaries may be required, although these can be unreliable.

Although there is an understanding that not all resources have been digitized, and that material artefacts are still extremely important as research objects in their own right, there is clearly frustration that online access, facilitated by seamless search, is not comprehensive and universal. There are issues around varying levels of **online access** to **digital** journals and databases caused by "*patchy institutional subscriptions*" (212) or as an outcome of being in the field or unaffiliated researcher status. This is compounded by problems with **search**. Within e-books or databases there is an expectation that full-text search is readily available (and fully functional) with high precision ("*There are a lot of bogus references to items .. that show up in search engines*" (301)) and recall ("*the database search was not picking up articles/reviews that I knew should be there*" (284)) anticipating user context: "*If one searches for 'organum' in the database 'Academic Search Premier' -- all sorts of medical journal articles pop up.*" (233).

Lack of **time** is a big problem for these participants, not to be wasted on "*learning software that I don't end up using*" (363). Infrequent use of complex software in research workflows leads to difficulties: "*Every time I come back to it, it feels like I have to learn it all over again*" (363).

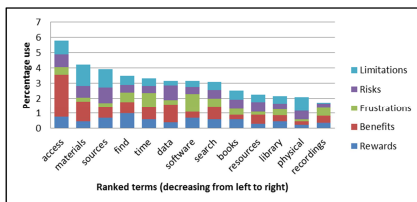


Figure 4. Key terms ranked by percentage use

4. DISCUSSION

On examination of the various pre *h*-point vocabulary analyses discussed above, it is clear that while the participants are enthusiastic about the rewards and benefits of the use of technology in their research, they have strong reservations around the risks and limitations of these technologies, which are often realized through frustrations when trying to achieve their research objectives. In particular the issues around **access**, **books** and **sources**, **finding** and **searching** and **time** are considered to be both positive and negative (Figure 4).

In Figure 5 the use of the key terms is broken up by digital skills of participants: the closer to the centre the line becomes, the less frequently the term is used. This data is incomplete (n=2 for DL1; n=18 for DL2) and un-

likely to be representative (reinforced by close examination of the terms in context) and is not included here. However it is interesting to observe that there appears to be more emphasis on technical terms (**software, data**) by DL5 while the least frequently used term by DL3 is **software**. **Librarians** are emphasized by DL3, while **sources** are ranked lower by DL5 than by their counterparts.

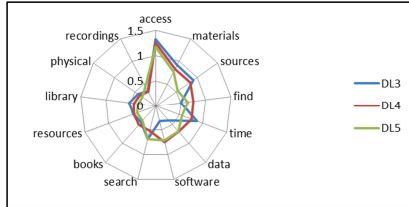


Figure 5. Key terms and digital skills ranked by percentage appearances in texts

The general consistency in the ranking of these terms is notable, reinforcing the idea that there is likely to be an agreed vocabulary and common practices within this community. Also, a common set of disciplinary values seem to emerge from the responses, emphasising qualities such as completeness, depth of analysis, accuracy, reliability, serendipity and the materiality of resources. It was observed above that musical resources were preferred by only a minority (7%) of the respondents. One possible explanation is that researchers study known musical items and mainly gather information *about* the music. However, many researchers study the musical content of archives and manuscript collections, and editing music is often their core activity. This relates in an interesting way to shortcomings that are observed in music printing software such as Finale and Sibelius, especially for creating scholarly editions of early music. Also, no tool support is reported for managing editorial data. There is clearly a case to be made for the development of systems that support the entire editorial workflow.

In summary, the (self-defined) musicologists who kindly took this survey and provided us with their thoughts clearly have access to technology (they did the survey online) and have positive and negative views (often held simultaneously) about its value in their research process. They may work unaffiliated and alone, or in an office with colleagues, and it is quite likely they are interested in historical or cultural musicology, or popular music studies. They are really excited about the increased access afforded by digital technologies and resources but some are wary of how digitization may make research superficial, undermining the discipline. They are habitual readers and want context-dependent access to physical and digital artefacts. They use software when it contributes to their workflow, and have a range of levels of digital fluency. Respondents rated their digital skills quite highly. However, the problems they report with consumer technologies suggest that they often overrated themselves. Also, there are many signs of insecurity in working with digital resources. Digital methodologies are ap-

parently not yet well integrated with mainstream research practice.

5. CONCLUSIONS AND FUTURE WORK

It is suggested that the research practices of musicologists reflect wider existing research into the digital humanities and that efforts should be made into supporting the development of their digital skills and in providing reliable user-centred software. This software should support online access to high quality digital resources (image, text, sound) which are comprehensive and discoverable, and can be shared, reused and manipulated at a micro- and macro level.

In the above we have presented an initial analysis of the WDMAD data, and while the size of the sample allows some generalization we recognize that there are likely to be differences amongst sub-disciplines within the population. Further work will examine the data at a more granular level, providing better understanding of work practices within sub-disciplines. Resources and software mentioned by participants also merit attention, for example for creating a collection of application scenarios. Finally, a comparison of the vocabularies used by musicologists and MIR researchers to describe technology may help to identify areas where misunderstanding may arise or values may clash. After completing this analysis, we will make the data available in a form that guarantees the anonymity of the participants.

Although it is too early to know in detail what musicologists do all day, we will use the findings of the WDMAD survey to guide the next steps in our research, which will include in-depth interviews, work with focus groups and co-design of prototype tools in the pursuit of answering this rather big question. Ultimately, we hope to raise the awareness of the importance of musicology centred design, and to contribute to the systematic creation of usable software and resources that enhance (and may ultimately transform) musicological research.

6. ACKNOWLEDGEMENTS

The authors would like to express their gratitude to the participants who freely gave their valuable contributions, the societies who helped by promoting the survey, and the colleagues who offered guidance on SPSS and translations. Frans Wiering was supported by the FES project COMMIT/ and the AHRC project Transforming Musicology.

7. REFERENCES

- [1] Benyon, D. (2014). *Designing Interactive Systems: A Comprehensive Guide to HCI, UX and Interaction Design*. Pearson.
- [2] Barrett, A. (2005). The information seeking habits of graduate student researchers in the humanities. *The Journal of Academic Librarianship* 31(4) : 324-331.
- [3] Bulger, M. & Meyer, E., De la Flor, G., Terras, M., Wyatt, S., Jirotko, M., Eccles, K., Madsen, C. (2011). Reinventing Research? Information Practices in the

- Humanities. A Research Information Network Report, April 2011. doi: 10.2139/ssrn.1859267
- [4] Gibbs, F. & Owens, T. (2012). Building Better Digital Humanities Tools: Toward broader audiences and user-centered designs. *Digital Humanities Quarterly* 6 (2) available online at <http://www.digitalhumanities.org/dhq/vol/6/2/000136/000136.html> [accessed 10 Oct 2014]
- [5] Schreibman, S. & Hanlon, A. (2010). Determining Value for Digital Humanities Tools: Report on a Survey of Tool Developers. *Digital Humanities Online* 4 (2) available online at <http://digitalhumanities.org/dhq/vol/4/2/000083/000083.html> [accessed 10 Oct 2014]
- [6] Warwick, C., Terras, M., Huntington, P., & Pappa, N. (2008). If you build it will they come? The LAIRAH study Quantifying the Use of Online Resources in the Arts and Humanities through Statistical Analysis of User Log Data. *Literary and Linguistic Computing* 23 (1) : 85-102.
- [7] Warwick, C. (2012). Studying users in digital humanities. In: Warwick, C., Terras, M., Nyhan, J. *Digital Humanities in Practice*. London: Facet Publishing, 2012. 1-21
- [8] Wiberley, S. & Jones, W. (1989). Patterns of information seeking in the humanities. *College & Research Libraries* 50 (6) 638 - 645
- [9] Lehman, S. & Renfro, P. (1991). Humanists and Electronic Information Services: Acceptance and Resistance. *College & Research Libraries* 52 (5) 409 - 413
- [10] JISC (2011) Developing Digital Literacies, available online at <http://www.jisc.ac.uk/media/documents/funding/2011/04/Briefingpaper.pdf> [accessed 15 Oct 2014]
- [11] Barthet, M. & Dixon, S. (2011). Ethnographic observations of musicologists at the British Library, *Proc. of the 12th International Society For Music Information Retrieval Conference*
- [12] Bonardi, A. (2000). IR for Contemporary Music: What the Musicologists Needs, *Proc. of the International Symposium on Music Information Retrieval*, 2000
- [13] Cook, N. (2005). Towards the compleat musicologist? *Proc. of the International Symposium on Music Information Retrieval*, 2005
- [14] Neubarth, K., Bergeron, M. & Conklin, D. (2011). Associations between musicology and music information retrieval. *Proc. of the 12th International Society For Music Information Retrieval Conference*
- [15] Lee, J. (2010). Analysis of user needs and information features in natural language queries seeking music information. *Journal of the American Society for Information Science and Technology*, 61(5), 1025-1045.
- [16] LaPlante, A. (2011). Social capital and music discovery: an examination of the ties through which late adolescents discover new music. *Proc. of 12th International Society for Music Information Retrieval Conference*.
- [17] Liew, C. & Ng, S. (2006). Beyond the Notes: A Qualitative Study of the Information-Seeking Behavior of Ethnomusicologists. *Journal of Academic Librarianship*, 32 (1) 66-68
- [18] Inskip, C., MacFarlane, A., Rafferty, P. (2010). Creative professional users musical relevance criteria. *Journal of Information Science* 36 (4), 517-529.
- [19] Weigl, D. & Guastavino, C. (2011). User studies in the music information retrieval literature. *Proc. of the 12th International Society For Music Information Retrieval Conference*
- [20] Borgman, C. (2009). The Digital Future is Now: A Call to Action for the Humanities. *Digital Humanities Quarterly* 3 (4) available online at <http://www.digitalhumanities.org/dhq/vol/3/4/000077/000077.html> [accessed 10 Oct 2014]
- [21] Transforming Musicology (2014). Transforming musicology. Available online at <http://www.transforming-musicology.org/> [accessed 13 Oct 2014]
- [22] European Science Foundation (2012). Musicology (Re)-mapped. Standing Committee for the Humanities: Discussion Paper. Available online at http://www.esf.org/fileadmin/Public_documents/Publications/musicology.pdf [accessed 10 Oct 2014]
- [23] Wiering, F. (2013). User needs and challenges in digital musicology. Available online at <http://www.staff.science.uu.nl/~wieri103/presentations/WieringLondonDigitalMusicLabFinal.pdf> [accessed 10 Oct 2014]
- [24] Volk, A. & Wiering, F. (2011). Musicology. In *Proc. of the Twelfth International Society for Music Information Retrieval Conference* (pp. 18). University of Miami. Presentation available online at <http://ismir2011.ismir.net/tutorials/ISMIR2011-Tutorial-Musicology.pdf> [accessed 17 Oct 2014]
- [25] Next History (n.d.). Enhancing Historical Research with Text Analysis Tools. Available online at <http://nexthistory.org/> [accessed 10 Oct 2014]
- [26] Popescu, I.-I. (2011). Text ranking by the weight of highly frequent words, in Grzybek, P. (Ed.) & Köhler, R. (Ed.) (2011). *Exact Methods in the Study of Language and Text*. Berlin, Boston: De Gruyter Mouton.
- [27] Hirsch, J. E. (2005). An index to quantify an individual's scientific research output. *Proceedings of the National Academy of Sciences of the United States of America*, 102(46), 16569-16572. doi:10.1073/pnas.0507655102

COMBINING FEATURES FOR COVER SONG IDENTIFICATION

Julien Osmalskyj

University of Liège
Belgium

josmalsky@ulg.ac.be

Peter Foster, Simon Dixon

Queen Mary University of London
United Kingdom

{p.a.foster, s.e.dixon}@qmul.ac.uk

Jean-Jacques Embrechts

University of Liège
Belgium

jjembrechts@ulg.ac.be

ABSTRACT

In this paper, we evaluate a set of methods for combining features for cover song identification. We first create multiple classifiers based on global tempo, duration, loudness, beats and chroma average features, training a random forest for each feature. Subsequently, we evaluate standard combination rules for merging these single classifiers into a composite classifier based on global features. We further obtain two higher level classifiers based on chroma features: one based on comparing histograms of quantized chroma features, and a second one based on computing cross-correlations between sequences of chroma features, to account for temporal information. For combining the latter chroma-based classifiers with the composite classifier based on global features, we use standard rank aggregation methods adapted from the information retrieval literature. We evaluate performance with the Second Hand Song dataset, where we quantify performance using multiple statistics. We observe that each combination rule outperforms single methods in terms of the total number of identified queries. Experiments with rank aggregation methods show an increase of up to 23.5 % of the number of identified queries, compared to single classifiers.

1. INTRODUCTION

Recent years have seen an increased interest in cover song recognition problems in the Music Information Retrieval (MIR) community. Such systems deal with the problem of retrieving different versions of a known audio query, where a version can be described as a new performance or recording of a previously recorded track [26]. Cover song recognition is a challenging task because the different renditions of a song may differ from the original work in terms of tempo, pitch, instrumentation or singing style. It is therefore an ongoing challenge to design features which are robust to variation in these musical characteristics.

Several approaches have been studied for cover song recognition problems. In existing work, retrieving cov-

ers is usually done by performing pairwise comparisons between audio queries and a reference database [10, 13, 14, 26], or by using index-based methods [2, 3, 16, 18]. A comprehensive review of existing methods is given in [24]. All these methods are based on single chroma representation, and do not consider using multiple features. Only few authors have considered the combination of features and distance measures. In the work of Foster et al. [11], multiple chroma-based distances are computed, then combined after ranking distances. Similarly, in an investigation performed by Ravuri et al. [22], the authors compute multiple chroma-based input features at multiple time scales, and combine them using a linear model. Finally, authors in Osmalskyj et al. [20] compare a range of methods for combining multiple spectral features for cover song identification.

In this paper, we make a distinction between cover song retrieval and cover song identification. In the first case, given an audio query, the goal is to retrieve as many covers as possible in a database. In the second case, the goal is to extract some information about the query, similarly to what fingerprinting systems do [27]. In that case, it is sufficient to retrieve only one version of the requested song as a human listener will act as the final expert by confirming a match in the returned set of results. Cover song identification covers a different set of applications, such as identification of live music, query by example, or retrieving any information related to an unknown version.

To take into account multiple sources of musical information, we propose to process an audio query using several methods based on different features. First, supervised machine learning is used to build classifiers that return probability estimates of similarity based on global features, including the tempo, the duration, the loudness, the number of beats and the average chroma features. We then merge these classifiers using standard probabilistic fusion rules to build a composite classifier. Then, we combine the latter with two methods based on chroma features. The first one is based on comparing histograms of quantized chroma features, to take into account the harmonic content of the songs. The second one is based on the cross-correlation of chroma sequences and further accounts for temporal information. As the scores returned by all these methods have different scales, we propose to combine them at the rank level using standard rank aggregation techniques inspired by the information retrieval literature, especially techniques used in web search engines [9, 21, 23]. We



© Julien Osmalskyj, Peter Foster, Simon Dixon, Jean-Jacques Embrechts.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Julien Osmalskyj, Peter Foster, Simon Dixon, Jean-Jacques Embrechts. "Combining Features for Cover Song Identification", 16th International Society for Music Information Retrieval Conference, 2015.

demonstrate that combining global features with chroma based features for cover identification improves the results over methods based on single features.

The remaining of this paper is organized as follows. Section 2 gives an overview of our approach and describes our methodology. Section 3 details the combination rules evaluated throughout this research. In Section 4, we describe our experimental setup as well as the evaluation procedure. Section 5 presents the realized experiments and the results obtained. Finally, Section 6 concludes the paper.

2. APPROACH OVERVIEW

Cover songs are different versions of underlying original works. The notion of cover therefore closely relates to musical similarity between two songs. A cover song identification system may therefore be conceived as measuring the similarity between two songs to classify them into a *similar* or a *dissimilar* class. We consider a binary notion of cover song identity. Our approach is based on several pairwise comparison functions called *rejectors*, as used in [19]. A rejector is a function \mathcal{R} that takes two audio tracks as an input and returns a score ranking the similarity between two tracks. In a cover song identification scenario, one track is the query while the other one is any track of the database. Rejectors aim to filter out result candidates, while retaining a subset of the database containing at least one match with respect to the query.

We design several rejectors based on different features and combine them such that the global output takes the information brought by each rejector into account. We make the assumption that the outputs of rejectors based on different features are independent, and therefore contribute to improving the performance of the system. We first design multiple probabilistic rejectors based on several global features using random forests [5]. We next design a rejector based on the quantization of chroma features. Finally, to take into account temporal information, we implement a rejector that computes cross-correlations between sequences of beat synchronous chroma features. This technique was first proposed by Ellis et al. [10] and is used as a baseline in our research.

2.1 Probabilistic Rejectors

Previous work, performed by Osmalskyj et al. [19,20], demonstrates that features such as tempo, duration, or spectral features perform better than random. However, as such features are global and low-dimensional, they do not bring much information when taken individually. Based on that observation, we select several of these global features and combine them in order to build a composite classifier that takes advantage of each single feature. For each feature, we build a probabilistic rejector using supervised machine learning. To determine the similarity of candidates with respect to a query, we perform pairwise comparisons using the rejectors. Features are extracted from the tracks and used as an input for the learned model to predict a probability. The probabilistic rejectors are furthermore combined

using several rules to build a composite rejector.

2.2 Codebook Rejector

To take into account the harmonic content of the songs, we build a rejector based on the quantization of chroma features. Similar features have been used in [19] and [11]. For each track, chroma features are mapped to specific codewords. A track is then represented by a histogram of the frequency of each codeword, known as a *bag-of-features* representation [12]. Codewords are determined using an unsupervised K-Means clustering of 200,000 beat synchronous and unit-normalized chroma vectors. We evaluated the number of codewords in the range 25 to 100. Best performance was achieved with a clustering of 100 codewords. To account for key transpositions, we make use of the *Optimal Transposition Index* (OTI) [25] as it is a straightforward approach that has been used in many other investigations [1, 11, 19, 24].

The similarity between two bag-of-features representations is computed as the *cosine similarity* between both histograms. We evaluated the cosine similarity against Euclidean and Bhattacharyya distances, as well as a supervised learning based distance. However, best results were achieved with the cosine similarity. Furthermore, the cosine similarity is fast to compute, especially when the input vectors are normalized to unit norm, as it can be computed as a simple dot product.

2.3 Cross Correlation Rejector

To take into account temporal information, we implement a baseline algorithm, initially proposed by Ellis et al. in [10]. In that method, songs are represented by beat-synchronous chroma matrices. Beat-tracking is used to align chromas on detected beats. Comparing songs is then performed by cross-correlating entire chroma-by-beat matrices. Sharp peaks in the resulting signal indicate a good alignment between the tracks. The input chroma matrices are high-pass filtered along time. We re-implemented existing work using a high-pass filter with the *alpha* coefficient set to 0.99. To compute the cross-correlation, we used a 2-dimensional FFT. This, on one hand, allows to find the optimal lag in the time dimension, and on the other hand, to find the best transposition shift along the chroma pitches. To emphasize sharp local maxima, the resulting cross-correlation signal is high-pass filtered. The final distance between two songs is taken as the reciprocal of the peak value of the cross-correlated signal.

3. COMBINING REJECTORS

The core of our method lies in the combination of rejectors. We first build probabilistic rejectors based on global features and combine them to produce a composite rejector. We evaluate several probabilistic fusion rules. Then, we combine that composite rejector with two other rejectors based on chroma features, using rank aggregation methods. This section details both kinds of combinations.

3.1 Score-based Combination

As stated in Section 2.1, previous work shows that rejectors based on global features such as the tempo or the duration of the songs do not produce satisfying results, when taken individually. It makes therefore sense to investigate their combination so that more information is taken into account when comparing two songs. As the global rejectors estimate probabilities of cover identities, we evaluate several combination rules to take advantage of each feature. Multiple rules have been proposed as a mean of combining probability estimates for classification [7, 8, 15]. We select in particular the *product*, the *sum* and the *median* rules [15] and evaluate the combination of our probabilistic rejectors with them.

3.1.1 Product Rule

The probabilistic product decision rule combines the a posteriori probabilities generated by the individual rejectors by a product rule. For N rejectors, the rule is given by

$$p = \frac{\frac{1}{C_s^{N-1}} \prod_{j=1}^N R_{j,s}}{\frac{1}{C_s^{N-1}} \prod_{j=1}^N R_{j,s} + \frac{1}{C_d^{N-1}} \prod_{j=1}^N R_{j,d}} \quad (1)$$

where C_s is the a priori probability of the similar class, C_d is the a priori probability of the dissimilar class, and $R_{j,s}$ (respectively $R_{j,d}$) is the probability that the rejector R_j considers the input tracks similar (respectively dissimilar). According to [15], it is a severe rule as it is sufficient for one rejector to inhibit a particular interpretation by outputting a close to zero probability for it.

3.1.2 Sum Rule

The sum probabilistic rule computes the final probability by computing the sum of each probability and averaging it by the number of rejectors. It is expressed as

$$p = \frac{1}{N} \sum_{i=1}^N R_j \quad (2)$$

where N is the number of rejectors and R_j is the probability returned by rejector j . For a set of classifiers that show independent noise behavior (e.g. based on different sets of features), the errors in the probability estimates are averaged by the summation [7]. In particular, the sum rule can be useful in reducing the noise for large sets of classifiers.

3.1.3 Median Rule

The median probabilistic rule is computed by taking the median of the individual probabilities. It is well established that the median is a robust estimate of the mean. The probabilistic sum in Equation 2 computes the average of the a posteriori probabilities. Therefore, if one rejector outputs an outlier probability, it will affect the final probability and it could lead to an incorrect decision. In that case, it might be more appropriate to use the median rule rather than the sum rule [15].

3.2 Rank Aggregation

While the composite global rejectors built by probabilistic fusion rules output probability estimates, two remaining rejectors, based on chroma features, return scores on different scales. Consequently, the rules described in Section 3.1 do not apply for fusing all rejectors together. As each rejector returns a list of ordered tracks, we propose to fuse all rejectors based on rank aggregation techniques, adapted from the information retrieval literature. Rank aggregation methods have been particularly studied in the web literature [9, 21, 23]. Compared to score-based combination, rank-aggregation is more suited as it is naturally calibrated and scale insensitive [21]. Indeed, using the returned scores requires to rescale the score values to the same range (e.g. between 0 and 1) so that different scales do not influence the aggregation results. Another advantage of rank aggregation is that the methods are usually computationally cheap as they usually consist in arithmetic operations on integer ranks. Furthermore, they require none or few parameters to set up.

In the case of cover song identification, each rejector compares queries to the entire search database and returns a full permutation of the database. Rank aggregation methods look at the position of each track in each list, and compute an aggregated rank to be associated to each track in the final list. A new list of results is then built by setting each track at the new rank position. We evaluated three rank aggregation rules: *minimum rank*, *mean rank*, *median rank*. For each track, we retrieve its rank in each input list, which allows us to aggregate ranks by respectively computing the minimum, the mean and the median of the ranks for each track. The final aggregated list is then sorted according to the new rank. Details of the experiments and the results are given in Section 5.

4. EXPERIMENTAL SETUP

4.1 Evaluation Database

For evaluation, we use the Second Hand Song dataset¹ (SHS), which is a subset of the Million Song Dataset [4] (MSD). The SHS is organized into 5,854 *cliques*, which correspond to groups of cover songs of original works. It contains on average 3,097 versions for 5,854 original songs. The SHS does not provide audio files, but contains pre-computed features such as the tempo, the duration, the beats, the loudness and the chroma features for 18,196 tracks, which makes it suitable for our research. Furthermore, it has been used in several research papers [3, 13, 14, 18], which allows us to compare our results to other methods.

The SHS proposes a pre-defined learning set (LS) and test set (TS), respectively containing 70% (12,960 tracks) and 30% (5,236 tracks) of the samples. However, to evaluate our method with variable LS and TS sizes, we merged both provided sets into one large set of 18,196 songs so that we can split it to different LS and TS sizes. Typically, since

¹ <http://labrosa.ee.columbia.edu/millionsong/secondhand>

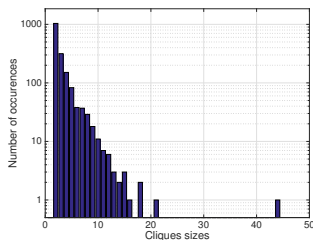


Figure 1: Distribution of the size of the cliques in the SHS dataset. Most of the cliques have a constant size of 2 or 3. However, some cliques contain more elements. The evaluation is therefore specific to that dataset as songs containing more versions will be more likely to be identified.

supervised learning algorithms such as the random forests require a decent amount of training samples, we set the LS to 70% and the TS to 30% of the SHS. However, to investigate how the system behaves on a larger scale, we also experimented with a larger TS containing 10,870 tracks. As the SHS provides a list of known duplicate tracks, we removed them from the dataset. Note that due to the removal of the duplicates, the number of cliques is reduced to 5,828, losing 26 cliques in the process.

It should be noted that the cliques in the SHS do not have a constant size, as can be seen in Figure 1. Although most of the cliques contain two elements, some cliques contain a lot more cover versions. Such songs containing many cover versions will be more likely to be identified in that evaluation set. The interpretation of the evaluated metrics remains therefore limited to the SHS dataset, as they characterize not only the identification algorithm, but also the dataset used to assess them.

4.2 Rejectors

Each rejector described in Section 2 makes use of the features pre-computed in the SHS. We specifically make use of the tempo, the duration, the loudness, the beats as well as the chroma features. The chroma features provided in the SHS are aligned on onsets rather than on the beats. As our chroma rejectors make use of beat-synchronous chroma features, we aligned the provided chromas on the provided beats, therefore approaching the beat-aligned representation proposed in Ellis et al. [10]. Note that in the work of Khadkevich et al. [14], the authors computed their own chroma features and compared them to the ones provided in the SHS. They report an improvement of 9.87% in terms of mean average precision against the chromas provided in the SHS with their chroma extraction algorithm. We therefore expect our method to perform better using a different chroma implementation (compared to the results presented in Section 5).

To account for differences in key for our probabilistic rejector based on average chroma features, we compute the

OTI [25] between average chromas and shift one chroma accordingly, similarly to what is done in Section 2.2 with the codebook rejector.

For the random forest algorithm, we use both a LS containing 70% of the cliques (selected at random) of the SHS, and a LS containing 40% of the cliques to study how the system behaves on a larger scale. A model is learned for each feature by processing the samples of the learning set. Note that to avoid overfitting during the learning phase, the depth of the trees is limited and the optimal depth is found by maximizing the area under the *Receiver Operating Characteristic* (ROC). The models are learned with 100 trees and with a maximal depth of 11.

4.3 Evaluation Algorithm and Metrics

For evaluation, each track of the TS is taken as a query and compared to the remaining tracks of the TS using our rejectors. As the results are provided for each query as a list of tracks ordered by descending order of similarity, we compute scores such as the Mean Rank (MR) of the first identified cover, the Mean Reciprocal Rank (MRR) and the Mean Average Precision (MAP) [17]. The MR corresponds to the mean position of the first identified query (lower is better). The MRR is computed as the average of the reciprocal of the rank of the first identified query (higher is better). The MAP for a set of queries corresponds to the mean of the average precision scores for each query (higher is better). Note that since we are interested in cover song identification rather than retrieval, we are only interested in retrieving at least one match for each query. Therefore, MR and MRR are more suited than the MAP as the latter takes into account the position of all matches in the list of results and is therefore only given as indicator. We also report the results in terms of the number of queries identified in *top-k* position, with *k* set to 10, 100 and 1000. This metric is also used in the MIREX evaluation [6].

5. RESULTS

5.1 Combining global rejectors

To investigate the behavior of probabilistic combination rules, as presented in Section 3.1, we combined our probabilistic rejectors based on global features using the product rule, the sum rule and the median rule. We first analyzed how each single rejector behaves on an evaluation database containing 5,464 tracks, compared to random classification. For the latter case, we simply built a rejector that outputs a probability sampled at random from a uniform distribution. Figure 2 shows curves corresponding to each rejector. Examination of the curves of the single rejectors shows that the rejector based on average chroma features performs better than the others (+92.5 % for top-10 and +18.5 % for top-100 compared to tempo). The tempo, beats and duration rejectors have similar curve shapes and perform similarly when taken individually. The composite median rule (in dark bold), obtained by fusing all single rejectors using the rule described in Section 3.1.3, performs better than the individual rejectors. In terms of the number

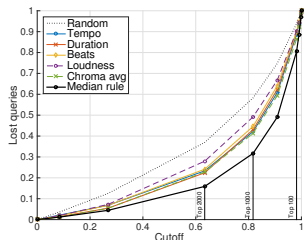


Figure 2: Single rejectors based on global features and composite rejector resulting from the probabilistic median combination rule, with an evaluation set of 5,464 tracks. The composite rejector outperforms any single rejector.

of tracks identified in the top-10, 100 and 1000, there is an improvement of respectively 62.5 %, 43.7 % and 16.4 %, compared to the average chroma rejector. In terms of MR and MRR, the composite rejector improves the scores by 24.9 % and 63.2 % respectively. To establish how all combination rules behave, Figure 3 displays the curves corresponding to each rule. Overall, all rules behave similarly. Zooming in the lower left corner (higher cutoff), the sum rule outperforms the product and the median. Compared to the median, the number of tracks identified in the top-5000 (lower-left area) is increased by 0.39% (5,419 tracks over 5,398). Similarly, the product rule outperforms other rules in the upper right corner (lower cutoff), with an increase of 24 % and 4 % for the top-10 and top-100 over the median rule. Our final choice is the median rule, as it produces a MR of 979.6 compared to 1127 and 1090 with the sum and product rules respectively.

5.2 Rank aggregation results

We combined the composite rejector based on global features with chroma based rejectors based on the quantization of chroma features and based on the cross correlation of chroma sequences. The three rank aggregation methods described in Section 3.2 are evaluated. We first report the results on a TS containing 30% of the SHS samples containing 1,745 cliques and 5,464 queries. Table 1 shows the number of queries identified in the top 10, 100 and 1000 for each single rejector and for each aggregation rule. Examining the results, we observe that each aggregation rule outperforms each single rejector. Best results for the top-10 returned tracks are achieved with the minimum aggregation rule. The number of identified tracks in the top-10 goes from 871 with the cross correlation rejector to 1004 with the minimum rule, which corresponds to an improvement of 15.2%. Best results for the top-100 and top-1000 returned tracks are both achieved with the mean rule, with improvements of respectively 23.5 % and 7.19 %. Figure 4 shows the performance of the minimum rank aggregation rule against each single rejector. The zooms in the lower left and upper right corners indicate that the aggregated

Top	Proba	Cluster	XCorr	Min	Mean	Median
10	169	560	871	1004	972	916
100	1064	1731	1523	2042	2139	2113
1000	3732	3931	3386	4177	4214	4129

Table 1: Results for a TS of 1745 cliques and 5,464 tracks. Rank aggregation combinations increase the number of identified queries for each rule.

	Proba	Cluster	XCorr	Min	Mean	Median
MR	979.6	861.4	1166	718.3	704.3	749.5
MRR	0.016	0.059	0.122	0.107	0.112	0.104
MAP	0.008	0.027	0.067	0.055	0.059	0.054

Table 2: Results for a TS of 1745 cliques and 5,464. Each rank aggregation combination outperforms single rejectors in terms of the Mean Rank (MR).

rejector performs better across the whole range of cutoff values. We also report the standard metrics (described in Section 4.3) in Table 2. Surprisingly, the MRR and MAP values are slightly decreased when compared to the best performing single rejector (cross-correlation, XCorr in the table). This might be due to the fact that when we aggregate the lists of results (Section 3.2), several tracks can be ranked at the same position. This might therefore affect the metrics. Note however that in terms of the Mean Rank, each combination outperforms each single rejector.

To establish how the aggregated rejectors scale on a larger dataset, we evaluated it on a TS containing 60% of the samples of the SHS. The LS used for learning the probabilistic rejectors is therefore smaller (40%) and produces decreased performance for the machine learning models built with random forests. That new TS contains 10,870 tracks, and is chosen to approach the size of the original SHS training set (12,960 tracks), to compare our results to results proposed in existing research papers [2, 13, 14]. We further increased the size of the TS by decreasing the size of the LS to 30% and 20% of the SHS. However, the produced results with the probabilistic rejectors showed worse performance, due to the lack of enough learning samples for the random forest algorithm. Table 3 shows the results of our method against existing work. Note that care should be taken while reading these results as our probabilistic models do not perform as well as with a larger LS, and as the sizes and the contents of both evaluation databases differ. In terms of the MR, our method is ranked at the second position.

6. CONCLUSION

In this paper, we evaluated multiple techniques for combining distances and features for cover song identification. We first made use of random forests to design probabilistic rejectors based on global features. We evaluated several standard combination rules such as the sum, the product and the median rules to build a composite rejector. Results show that combining single rejectors based on global fea-

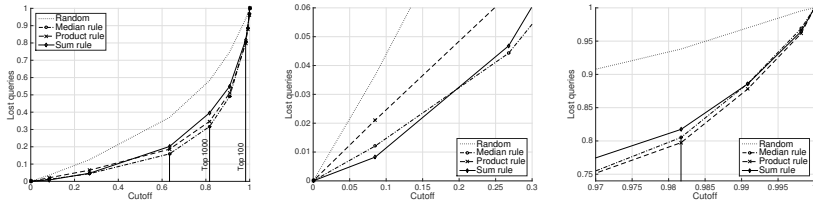


Figure 3: Performance of the probabilistic sum, product and median combination rules to build a composite rejector based on multiple global features. The second figure is a zoom of the left lower part (high cutoff). The sum rule performs slightly better in that area. The third figure is a zoom of the upper right area. The product rule performs slightly better there.

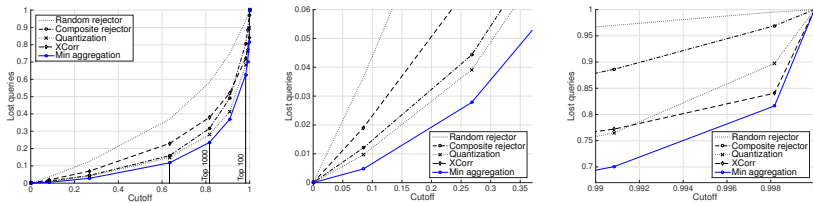


Figure 4: Performance of the minimum aggregation rule against rejectors based on global features (composite), quantization of chroma features and cross-correlation of chroma sequences (XCorr) on a database containing 5,464 tracks. The second figure is a zoom of the lower left corner (high cutoff) and the third figure is a zoom of the upper right corner (low cutoff). In each case, the aggregation increases the number of identified tracks.

Method	MR	MAP
Khadkevich et al. [14]	958.2	0.10
Rank Aggregation (10,870 tracks)	1,455.6	0.048
Bertin-Mahieux et al. 2D-FTM (200 pcs) [3]	3,005	0.09
Humphrey et al. [13]	1,844	0.28

Table 3: Comparison of the rank aggregation method against existing methods evaluated on the SHS original training set. Care should be taken when reading the results as the original SHS training set contains 12,960 songs, and our subset contains 10,870 tracks sampled from the SHS.

tures improves the performance compared to single classifiers. We proposed to combine the composite rejector based on global features with rejectors based on chroma features. To take into account the harmonic content of the songs, we introduced a rejector based on comparing histograms of quantized chroma features. To account for temporal information, we further implemented a baseline rejector performing cross-correlations between sequences of chroma features. As all these rejectors return values on different scales, we proposed to combine them at the rank level. We evaluated several rank aggregation methods such as the mean, the median and the minimum aggregation rules. We conducted experiments on the Second

Hand Song dataset and observed that aggregation methods outperform methods in isolation for cover song identification. Results are provided in terms of standard metrics such as the mean rank of the first match, the mean reciprocal rank and the mean average precision, as well as in terms of the total number of queries identified in the top-k results. Compared to single rejectors, the minimum aggregation rule shows an improvement of up to 23.5 % of the number of queries identified in the top-100 returned tracks. Comparing our results to existing work, we observe that our method does not perform as well as other methods in terms of mean average precision. However, in terms of mean rank of the first identified query, the results are comparable to related methods and rank our method at the second position. Although our method does not produce state-of-the-art results, we showed that aggregating multiple features and distance measures does increase the number of identified queries. These results suggest that combining many other features as well as multiple comparison algorithms could lead to significant improvements in any cover song identification system. Future work therefore includes more experiments with features taking into account e.g. the melodic line of the songs, or structural information. In any case, many combining experiments should still be performed to improve state-of-the-art results.

7. REFERENCES

- [1] T. Ahonen. Compression-based clustering of chromagram data: New method and representations. In *International Symposium on Computer Music Modeling and Retrieval*, pages 474–481, 2012.
- [2] T. Bertin-Mahieux and D. Ellis. Large-scale cover song recognition using hashed chroma landmarks. In *Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, 2011.
- [3] T. Bertin-Mahieux and D. Ellis. Large-scale cover song recognition using the 2D Fourier transform magnitude. In *Proceedings of the 13th International Conference on Music Information Retrieval (ISMIR 2012)*, 2012.
- [4] T. Bertin-Mahieux, D. Ellis, B. Whitman, and P. Lamere. The million song dataset. In *Int. Symp. Music Inform. Retrieval (ISMIR)*, 2011.
- [5] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Jan. 2001.
- [6] J. Downie, A. Ehmann, M. Bay, and M. Jones. The music information retrieval evaluation exchange: Some observations and insights. In *Advances in music information retrieval*, pages 93–115. Springer, 2010.
- [7] R. Duin. The combining classifier: to train or not to train? In *IEEE Int. Conf. Pattern Recognition (ICPR)*, volume 2, pages 765–770, Quebec City, Canada, Aug. 2002.
- [8] R. Duin and D. Tax. Experiments with classifier combining rules. In *Multiple Classifier Systems*, volume 1857 of *Lecture Notes in Comp. Science*, pages 16–29. Springer, 2000.
- [9] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th international conference on World Wide Web*, pages 613–622. ACM, 2001.
- [10] D. Ellis and G. Poliner. Identifying cover songs with chroma features and dynamic programming beat tracking. In *Int. Conf. Acoustics, Speech and Signal Process. (ICASSP)*, volume 4, 2007.
- [11] P. Foster, S. Dixon, and A. Klapuri. Identifying cover songs using information-theoretic measures of similarity. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 23(6):993–1005, June 2015.
- [12] Z. Fu, G. Lu, K. Ming Ting, and D. Zhang. Music classification via the bag-of-features approach. *Pattern Recognition Letters*, 32(14):1768 – 1777, 2011.
- [13] E. Humphrey, O. Nieto, and J. Bello. Data driven and discriminative projections for large-scale cover song identification. In *Int. Symp. Music Inform. Retrieval (ISMIR)*, 2013.
- [14] M. Khadkevich and M. Omologo. Large-scale cover song identification using chord profiles. In *Int. Symp. Music Inform. Retrieval (ISMIR)*, pages 233–238, 2013.
- [15] J. Kittler, M. Hatef, R. Duin, and J. Matas. On combining classifiers. *IEEE Trans. Pattern Anal. Mach. Intell.*, 20(3):226–239, Mar. 1998.
- [16] F. Kurth and M. Muller. Efficient index-based audio matching. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):382–395, 2008.
- [17] C. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.
- [18] B. Martin, D. Brown, P. Hanna, and P. Ferraro. Blast for audio sequences alignment: A fast scalable cover identification tool. In *ISMIR*, pages 529–534, 2012.
- [19] J. Osmalskyj, S. Piérard, M. Van Droogenbroeck, and J.-J. Embrechts. Efficient database pruning for large-scale cover song recognition. In *Int. Conf. Acoustics, Speech and Signal Process. (ICASSP)*, pages 714–718, Vancouver, Canada, May 2013.
- [20] J. Osmalskyj, M. Van Droogenbroeck, and J.-J. Embrechts. Performances of low-level audio classifiers for large-scale music similarity. In *International Conference on Systems, Signals and Image Processing (IWS-SIP)*, pages 91–94, Dubrovnik, Croatia, May 2014.
- [21] R. Prati. Combining feature ranking algorithms through rank aggregation. In *Neural Networks (IJCNN), The 2012 International Joint Conference on*, pages 1–8. IEEE, 2012.
- [22] S. Ravuri and D. Ellis. Cover song detection: from high scores to general classification. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 65–68. IEEE, 2010.
- [23] D. Sculley. Rank aggregation for similar items. In *SDM*, pages 587–592. SIAM, 2006.
- [24] J. Serra. *Identification of versions of the same musical composition by processing audio descriptions*. PhD thesis, Universitat Pompeu Fabra, Barcelona, 2011.
- [25] J. Serra and E. Gómez. Audio cover song identification based on tonal sequence alignment. In *Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on*, pages 61–64. IEEE, 2008.
- [26] J. Serra, E. Gomez, P. Herrera, and X. Serra. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Trans. Audio, Speech and Language Process.*, 16(6):1138–1152, 2008.
- [27] A. Wang. An industrial-strength audio search algorithm. In *Int. Symp. Music Inform. Retrieval (ISMIR)*, pages 7–13, 2003.

SCORE FOLLOWING FOR PIANO PERFORMANCES WITH SUSTAIN-PEDAL EFFECTS

Bochen Li Zhiyao Duan

Audio Information Research (AIR) Lab,

University of Rochester, Department of Electrical and Computer Engineering

bli23@ur.rochester.edu, zhiyao.duan@rochester.edu

ABSTRACT

One challenge in score following (i.e., mapping audio frames to score positions in real time) for piano performances is the mismatch between audio and score caused by the usage of the sustain pedal. When the pedal is pressed, notes played will continue to sound until the string vibration naturally ceases. This makes the notes longer than their notated lengths and overlap with later notes. In this paper, we propose an approach to address this problem. Given that the most competitive wrong score positions for each audio frame are the ones before the correct position due to the sustained sounds, we remove partials of sustained notes and only retain partials of “new notes” in the audio representation. This operation reduces sustain-pedal effects by weakening the match between the audio frame and previous wrong score positions, hence encourages the system to align to the correct score position. We implement this idea based on a state-of-the-art score following framework. Experiments on synthetic and real piano performances from the MAPS dataset show significant improvements on both alignment accuracy and robustness.

1. INTRODUCTION

1.1 Audio-Score Alignment

Audio-score alignment is the problem of aligning (synchronizing) a music audio performance with its score [8]. It can be addressed either offline or online. Offline algorithms may “look into the future” when aligning the current audio frame to the score. Online algorithms (also called *score following*), on the other hand, may only use the past and current audio data to align the current audio frame to the score. Provided with enough computational resources, online algorithms can be applied in real-time scenarios. As online algorithms utilize less input data than offline algorithms, they can support broader applications including those in offline scenarios. However, they are also more challenging to achieve the same alignment accuracy and robustness as offline algorithms do.

Audio-score alignment has many existing and potential applications. Offline algorithms have been used for audio indexing to synchronize multiple modalities (video, audio, score, etc.) of music to build a digital library [28]. Other applications include a piano pedagogical system [3] and an intelligent audio content editor [11]. Online algorithms further support online or even real-time applications, including automatic accompaniment of a soloist’s performance [8], automatic coordination of audio-visual equipment [18], real-time score-informed source separation and remixing [11], and automatic page turning for musicians [1]. Potential applications of audio-score alignment include musicological comparison of different versions of musical performances, automatic lyrics display, and stage light/camera management.


1.2 Related Work

In this section, we briefly review existing approaches to audio-score alignment with an emphasis on score following for piano performances, which is the problem addressed in this paper.

Audio-score alignment has been an active research topic for two decades. Early researchers started with monophonic audio performances. Puckette [25], Grubb and Dannenberg [16], and Cano et al. [4] proposed systems to follow vocal performances. Orio and Dechelle [23] used a Hidden Markov Model (HMM)-based method to follow different monophonic instruments and voices. Raphael [26] applied a Bayesian network to follow and accompany a monophonic instrument soloist.

For polyphonic audio, a number of offline systems using Dynamic Time Warping (DTW) have been proposed for different kinds of instruments, including string and wind ensembles [24] and pop songs [17]. For online algorithms, Duan and Pardo [12] proposed a 2-dimensional state space model to follow an ensemble of string and wind instruments. All the abovementioned methods, however, have not been tested on piano performances.

There are a few systems that are capable of aligning piano performances. Joder and Schuller [20] proposed an HMM system with an adaptive-template-based observation model to follow piano performances. In [19], Joder et al. further improved the system by exploring different feature functions for the observation model and using a Conditional Random Field (CRF) as the alignment frame-

 © Bochen Li, Zhiyao Duan. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Bochen Li, Zhiyao Duan. “Score Following for Piano Performances with Sustain-Pedal Effects”, 16th International Society for Music Information Retrieval Conference, 2015.

work. Wang et al. [29] employed DTW to achieve alignment in three passes of the audio performance and used score-driven NMF to refine the audio and score representations in later passes. All the abovementioned systems have been systematically evaluated and shown with good performance on about 50 classical piano performances from the MAPS dataset [14], however, they are offline algorithms and require the entire audio piece to find the alignment. Dixo and Widmer [10] developed a toolkit to align different versions of music audio performances including piano based on an efficient DTW algorithm. However, this is again an offline algorithm, although an extension to online scenarios can be made through online DTW algorithms [9].

For online algorithms capable of following piano performances, Cont [5] proposed a hierarchical HMM approach with Nonnegative Matrix Factorization (NMF). However, this system was not quantitatively evaluated. Later, Cont [6] proposed another probabilistic inference framework with two coupled audio and tempo agents to follow general polyphonic performances. This algorithm has been systematically evaluated on 11 monophonic and lightly polyphonic pieces played by wind and string instruments, but just 1 polyphonic piano performance (a Fugue by J.S. Bach).

1.3 Our Contribution

In this paper, we are interested in following piano performances. Their specific properties, such as sustain pedal effects, the sympathetic vibration of strings, and the wide pitch range, may impose challenges to systems that are designed to follow ensembles of voices, strings, and wind instruments. In particular, we argue that the sustain-pedal effects are especially challenging. When the pedal is pressed, notes played will continue to sound until the string vibration naturally ceases. This makes the notes longer than their notated lengths and overlap with later notes, which causes potential mismatch between audio and score.

Note that Niedermayer et al. reported negligible influence of sustain-pedal effects on alignment results in their experimental study on audio-score alignment [22]. However, they further reasoned that this might be because the dataset used for evaluation contains only Mozart pieces, in which “the usage of pedals plays a relatively minor role”. In fact, the sustain pedal has been commonly used since the Romantic era (after Mozart) in the Western music history, and is widely used in modern piano performances of many different styles. Another reason for Niedermayer et al.’s observation, we argue, is that the algorithm used for evaluation was an offline algorithm, which is more robust to the local mismatch between audio and score as a global alignment is employed. For online algorithms, however, they are more sensitive to local audio-score mismatch and they can be totally lost during the following process.

In this paper, we build a system to follow piano performances, based on the state-space framework proposed by Duan and Pardo [12]. More specifically, we propose an approach to deal with the mismatch issue caused by sustain-pedal effects. In each inter-onset segment of the audio, we remove partials of all notes extended from the

previous segment and only retain partials of the new notes. This operation reduces sustain-pedal effects by weakening the match between an audio frame and the previous wrong score positions, which are the most competitive wrong candidates. But we need to mention another case that the match between this audio frame and the current correct score position may be also reduced, if notes in previous frames are actually extended because they are not released yet according to the score instead of due to the sustain pedal. Nevertheless, as explained in detail in Section 3.4, this operation still favors the correct position even in this case. We conduct experiments on 25 synthetic and 25 real piano performances randomly chosen from the MAPS dataset [14]. Results show that the proposed system significantly outperforms the baseline system [12] on both alignment accuracy and robustness.

2. SYSTEM FRAMEWORK

We build our system based on the state-space model proposed in [12], which follows polyphonic audio with its score. Music audio is segmented into time frames and fed into the system in sequence. Each frame \mathbf{y}_n is associated with a 2-dimensional state vector $\mathbf{s}_n = (x_n, v_n)^T$, representing its underlying score position (in beats) and tempo (in beats-per-minute), respectively. The goal of score following is to infer the score position x_n from current and previous audio observations $\mathbf{y}_1, \dots, \mathbf{y}_n$. This is formulated as an online inference problem of hidden states of a hidden Markov process, which is achieved through particle filtering. The hidden Markov process contains two parts: a process model and an observation model.

The process model describes state transition probabilities $p(\mathbf{s}_n | \mathbf{s}_{n-1})$ by two dynamic equations for x_n and v_n , respectively. The score position advances from the previous position according to the tempo. The tempo changes through a random walk or does not change at all, depending on where the position is.

The observation model $p(\mathbf{y}_n | \mathbf{s}_n)$ evaluates the match between an audio frame and the hypothesized state on the pitch content. A good match is achieved when the audio frame contains exactly the pitches described on the score at the hypothesized score position in the state. Otherwise, a bad match is achieved. This is calculated using the multi-pitch likelihood model proposed in [13], which evaluates the likelihood of a hypothesized pitch set in explaining the magnitude spectrum of an audio frame.

The multi-pitch likelihood model detects prominent peaks in the magnitude spectrum of the audio frame and represents them as frequency-amplitude pairs:

$$\mathcal{P} = \{(f_i, a_i)\}_{i=1}^K, \quad (1)$$

where K is the total number of peaks detected in the frame. The likelihood would be high if the harmonics of the hypothesized pitch set match well with the detected peaks in terms of both frequency and amplitude. The likelihood would be low otherwise, for example, if many harmonics are far away from any detected peak.

3. PROPOSED METHOD

3.1 Properties of Piano Music

There are many specific properties of piano music, such as the wide pitch range and the inharmonicity of note partials. In this section, we discuss two properties considered in the proposed approach: strong onset with exponential decay of the note waveform, and the sustain-pedal effects.

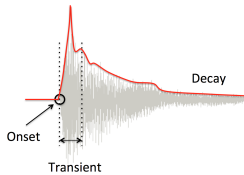


Figure 1. Waveform and energy envelope of a piano note.

Figure 1 shows the waveform and energy envelope of a piano note. We see a sudden energy increase at the onset followed by an exponential decay. When a piano key is pressed, its damper is released and its hammer strikes the strings, which yields an impulse-like articulation. The damper continues to be released as the key is being pressed. This lets the string vibration decay naturally, which may take as long as 10 seconds. The damper comes back to the strings when the key is released, and the string vibration ceases quickly. However, when the sustain pedal is pressed, all dampers of all keys are released no matter if a key is pressed or not. This allows all active notes to continue to sound, and even activate some inactive notes due to sympathetic vibrations, which enriches the sound timbre.

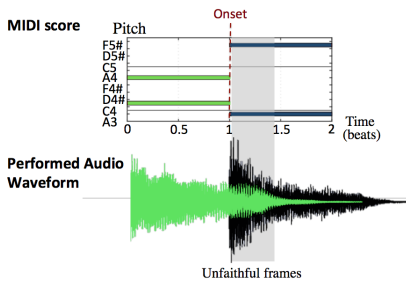


Figure 2. Mismatch between audio and score caused by the sustain-pedal effects.

A detailed analysis of the sustain-pedal effects is given by Lehtonen et al. in [21]. Here we focus on its resulted mismatch problem between audio and score. Figure 2 shows the MIDI score (in pianoroll) and waveforms of four notes. According to the score, the first two notes are supposed to end when the latter ones start. However, due to the sustain pedal, the waveforms of the first two notes are extended

into those of the latter. This causes potential mismatch between the audio and the score, especially in frames right after the onset of the latter notes. In other words, the audio is unfaithful to the score in those frames. The degree and the length of the unfaithfulness, however, is not notated in the score. It depends on the notes being played as well as how hard the performer presses the pedal. If the pedal is pressed partially, then the damper will slightly touch the strings and the effects are slighter. While some composers and music editors use pedal marks to notate it, appropriate use of the sustain pedal is more often left to the performer.

The main idea of the proposed approach to deal with the sustain-pedal effects is to first detect audio onsets to locate the potentially unfaithful frames. Then partials of the extended notes are removed in the peak representation of these frames. We describe the two steps in the following.

3.2 Onset Detection

Although not all frames right after an onset are unfaithful, as notes could be extended because their keys are still pressed according to the score, many unfaithful frames do appear right after onsets. Therefore, onset detection helps to locate potentially unfaithful frames. Many onset detection methods have been proposed in the literature [2]. In this paper, we adopt the widely used spectral-based approach, since it is effective for polyphonic signals. We adapt it to online scenarios for our score following system.

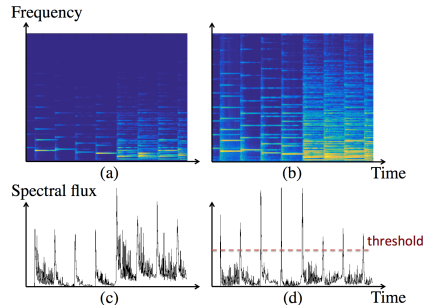


Figure 3. Illustration of onset detection. (a) Spectrogram. (b) Spectrogram after compression. (c) Spectral flux. (d) Normalized spectral flux by signal energy.

Figure 3 illustrates the onset detection process. We first calculate the audio magnitude spectrogram $\mathbf{Y}(n, k)$ through Short-time Fourier Transform (STFT) in Figure 3(a), where n and k are frame and frequency bin indices, respectively. We then apply logarithmic compression on it to enhance the high-frequency content by

$$\tilde{\mathbf{Y}}(n, k) = \log(1 + \gamma \cdot \mathbf{Y}(n, k)), \quad (2)$$

where γ controls the compression ratio. This is because high frequency content is indicative for onsets but relatively weak in the original spectrogram [27]. Figure 3(b)

shows the enhanced magnitude spectrogram with $\gamma = 0.2$. We then compute the spectral flux $\Delta_Y(n)$ by summing positive temporal differences across all frequency bins as

$$\Delta_Y(n) = \sum_k |\tilde{Y}(n, k) - \tilde{Y}(n-1, k)|_{>0}, \quad (3)$$

where $|\cdot|_{>0}$ denotes half-wave rectification, i.e., keeping non-negative values while setting negative values to 0. The calculated spectral flux is shown in Figure 3(c). We can see that all onsets in the example are associated with a clear peak, however, peak heights vary much. Spurious peaks in the middle of louder notes are as high as true peaks of softer notes. One could set an adaptive threshold which varies with the moving average of the spectral flux, but this would make the onset detection algorithm offline. Instead, we normalize the spectral flux by the energy of the audio signal in the current frame by

$$\hat{\Delta}_Y(n) = \Delta_Y(n)/E(n), \quad (4)$$

where $E(n)$ is the Root-Mean-Square (RMS) value of the n -th frame of the audio. After this operation, a simple threshold can detect the onsets, as shown in Figure 3(d).

Note that onset detection has been used in several online [5] and offline [15] alignment algorithms, where a special matching function is used to match audio and score onsets. In our system, however, onset detection is to locate potentially unfaithful audio frames. Their audio representations are modified but no special matching function is defined.

3.3 Reduce Pedal Effects by Spectral Peak Removal

Frames within a period after a detected onset are potentially unfaithful frames due to the sustain pedal. Conservatively, without knowledge of the degree and length of the effects, we just reduce them in the first 200ms (i.e., 20 frames) following an onset. As described in Section 2, each audio frame is represented by a set of significant spectral peaks in Eq. (1). The match between the audio frame and a hypothesized score location is evaluated through the multi-pitch likelihood model on how well the harmonics of the score notes match with spectral peaks in the audio. As the spectrum of an unfaithful audio frame contains unexpected peaks corresponding to partials of notes extended by the sustain pedal, we propose to remove these peaks to reduce the mismatch between audio and score.

Figure 4 illustrates the idea. For each potentially unfaithful frame (e.g., the n -th frame), we compare its spectral peaks with those in a frame before the onset (e.g., the m -th frame), and remove peaks that seem to be extended from the earlier frame. Let $\mathcal{P}_m = \{\langle f_i^m, a_i^m \rangle\}_{i=1}^{K_m}$ be the total K_m peaks detected in the m -th frame, and $\mathcal{P}_n = \{\langle f_j^n, a_j^n \rangle\}_{j=1}^{K_n}$ be the total K_n peaks detected in the n -th frame. A peak in the n -th frame whose frequency is very close to and whose amplitude is smaller than those of a peak in the m -th frame is considered as an extension and is removed. Note that repeated notes will not be removed in this way as the amplitude criterion is not met. Extended

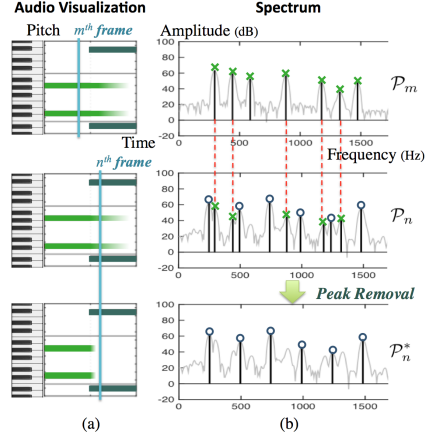


Figure 4. Illustration of the spectral peak removal idea. (a) Audio performance representation before and after peak removal. (b) Magnitude spectra with spectral peaks in the m -th and n -th frames. Peaks marked by crosses correspond to the first two notes. Peaks marked by circles correspond to the latter two notes.

partials that are overlapped with a partial of a new note will not be removed either due to the same reason. After peak removal, a new spectral peak representation of the n -th frame is obtained as

$$\mathcal{P}_n^* = \mathcal{P}_n - \{\langle f_i^n, a_i^n \rangle : \exists j \text{ s.t. } |f_i^n - f_j^m| < d, a_i^n < a_j^m\}, \quad (5)$$

where $\langle f_i^n, a_i^n \rangle \in \mathcal{P}_n$. d is the threshold for the allowable frequency deviation, which is set to a quarter tone in this paper. Finally, the match between the n -th frame and a hypothesized score position is evaluated through the multi-pitch likelihood of score-indicated pitches in explaining the modified peak representation of the spectrum. Note that this operation only modifies the peak representation of the audio instead of the audio itself.

The peak removal operation emphasizes new notes in the representation and discards old ones. This is in accordance to music perception, as we always pay more attention to new notes even though the old notes are as loud.

3.4 New Mismatch Introduced by Peak Removal

The peak removal operation removes notes extended by the sustain pedal in the audio representation, however, it also removes notes that should remain according to the score, e.g. D4 in Figure 5(a). This causes new mismatch between audio and score. Ideally, we could differentiate these two kinds of notes from the note offset information in a well-aligned score, which we do not have during score following. Nevertheless, we explain in the following that the new mismatch actually still helps with score following.

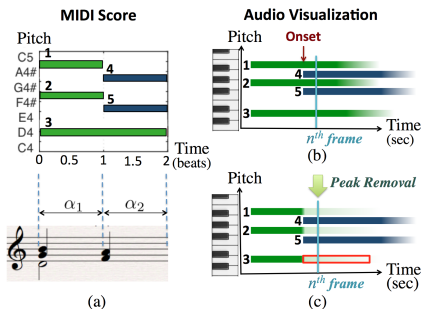


Figure 5. Illustration of mismatch reduced and introduced by the peak removal operation. (a) MIDI score and its piano-roll representation. (b) Audio performance representation before peak removal. (c) Audio performance representation after peak removal.

Figure 5 illustrates the mismatch reduced and introduced by the peak removal operation. A MIDI score with two inter-onset segments α_1 and α_2 is shown in Figure 5(a). Notes 1 and 2 are supposed to end when Notes 4 and 5 start, while Note 3 is supposed to span both segments. For an audio frame right after the onset (e.g., the n -th frame) in Figure 5(b), we can see that it contains all the five notes, including Notes 1 and 2 due to the sustain pedal. It is therefore unfaithful to the correct segment α_2 in the score. Which segment is a better match to this audio frame? Out of the 5 notes in the n -th frame, α_1 contains 3 (Notes 1, 2, and 3) and α_2 also contains 3 (Notes 4, 5, and 3). The correct segment α_2 does not show a better match than α_1 .

Suppose the audio onset of Note 4 and 5 is detected, then the peak removal operation will remove spectral peaks corresponding to Notes 1, 2, and 3 in the n -th frame. The mismatch between the n -th frame and the correct segment α_2 due to the sustain pedal is reduced, while new mismatch is introduced as Note 3 is supposed to stay in α_2 in the score but is removed in the audio. This leaves 2 notes (Notes 4 and 5) shared by the score and the audio, although the score has 1 more note (Note 3). The mismatch between the n -th frame and α_1 , on the other hand, is increased significantly. There becomes no intersection at all between notes remained in the n -th frame (Notes 4 and 5) and notes in α_1 (Notes 1, 2, and 3). Therefore, the correct segment α_2 is clearly a better match to the n -th frame.

In general, the peak removal operation may introduce mismatch between an audio frame and its correct score location as it may remove peaks that are supposed to stay, but the mismatch between the audio frame and the previous wrong score location will be increased much more. In fact, there will be no match at all. This is true even if all notes in α_1 stay in α_2 according to the score. Therefore, the mismatch introduced by the peak removal operation is not harmful to but actually helps with score following.

In Figure 5, we only consider the previous segment α_1

as a wrong segment to compete with α_2 . This is because it is the most common error caused by the sustain pedal in score following. The peak removal operation, however, can help eliminate non-immediate segments that are prior to the current segment as well.

4. EXPERIMENTS

4.1 Data Set and Evaluation Measures

We use the MAPS dataset [14] to evaluate the proposed approach. In this dataset, performers first play on a MIDI keyboard, then the MIDI performances are rendered into audio by a software synthesizer or a Yamaha Disklavier. The former are synthetic recordings while the latter are real acoustic recordings. Both have exactly the same timing as the MIDI performances. We randomly select 25 synthetic pieces and 25 real pieces from the dataset. The synthetic pieces simulate the “Bechstein D 280” piano in a concert hall, and the real pieces are recorded with an upright Disklavier piano. Approximately 18 synthetic pieces and 10 real pieces are played with substantial sustain pedal usage. We then download their MIDI scores from <http://piano-midi.de/>. Note that the MIDI performances have minor differences from the MIDI scores besides their tempo difference. These include occasionally missed or added notes, different renderings of trills, and slight desynchronization of simultaneous notes. We therefore perform an offline DTW algorithm to align the MIDI performances to the MIDI scores and then manually correct minor errors to obtain the ground-truth alignment.

We calculate the time deviation (in ms) between the ground-truth alignment and the system’s output alignment of the onset of each score note. This value ranges from 0ms to the total length of the audio. We define its average over all notes in a piece as the *Average Time Deviation (ATD)*.

We also calculate the *Align Rate (AR)* [7] for all pieces. It is defined as the percentage of correctly aligned notes, those whose time deviation is less than a threshold. Commonly used thresholds range from 50ms to 200ms depending on the application. For an automatic accompaniment system, a deviation less than 50ms would be required, while for an automatic page turner, 200ms would be fine.

4.2 Implementation Details

Our score following system is built upon the system proposed in [12], whose source code can be downloaded at the authors’ website. We therefore take it as the baseline system for comparison. We use the authors’ original code and parameter settings in both the baseline system and the proposed system. The multi-pitch likelihood model in [12] was trained on thousands of randomly mixed chords using notes of 16 kinds of Western instruments excluding piano. We stick with this model in the proposed system for a fair comparison. For unique parameters of the proposed system, we set γ to 0.2 in Eq. (2), the threshold in Figure 3 to 225, the length of unfaithful region to 200ms after each detected onset, the frame to compare with to the 5-th frame before the onset, and the peak frequency deviation d in Eq.

(5) to a quarter tone. All these parameters are fixed for all pieces. Due to the probabilistic nature of the baseline system and the proposed system, we run 10 times of each system on each piece for the comparison.

4.3 Results

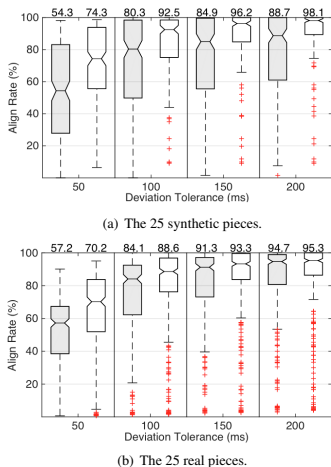


Figure 6. Align Rate comparisons between the baseline [12] (grey) and the proposed (white) systems using different time deviation tolerances. Numbers above the figures show medians of the boxes.

Figure 6 shows box plots of align rates of the two systems with different onset deviation tolerance values on both synthetic and real pieces. Each box in Figure 6(a) represents 250 data points (10 runs on 25 pieces) and each box in Figure 6(b) represents 250 data points. We can see that for the synthetic pieces, the median align rate is significantly improved for all tolerance values. The dispersion of the distribution is also significantly shrunk, making the improvement on some low-performing piece-runs especially significant. For the real pieces, the median align rate is significantly improved for all tolerance values except 200ms. The dispersion of the distribution is shrunk significantly for all tolerances except 50ms. This shows that the proposed approach improves the alignment accuracy and robustness significantly on both synthetic and real pieces. The improvement on synthetic pieces is more remarkable because there are more synthetic pieces with a substantial pedal usage. However, the proposed system also has more low-performing outliers on the real pieces, some of which correspond to piece-runs when the system is lost.

Figure 7 compares the Average Time Deviation (ATD) between the two systems on all piece-runs. Again, each box in the synthetic setting contains 250 points and each box in the real setting contains 250 points. We can see

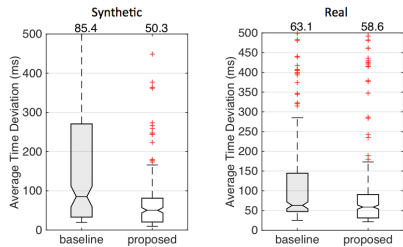


Figure 7. Average time deviation comparison between the baseline [12] and the proposed system. Outliers that exceed 500ms are not shown in this figure. Several outliers are higher than 3 seconds. Numbers above the figure show medians of the boxes.

that the median ATD in both cases is reduced by the proposed system. The reduction on the synthetic pieces is even more significant. The dispersion of the distribution is also shrunk significantly, reducing the worst ATD (excluding outliers) from 200-300ms to the range under 200ms. After the improvement, a fair amount of synthetic and real piece-runs have ATD under 50ms, which would enable real-time applications such as automatic accompaniment.

Examples of alignment results can be found at <http://www.ece.rochester.edu/users/bli123/projects/pianofollowing>.

5. CONCLUSIONS

In this paper we proposed an approach to follow piano performances with sustain-pedal effects. The usage of the sustain pedal extends notes even if their keys have been released, hence causes mismatch between audio and score, especially in frames right after note onsets. To address this problem, we first detect audio onsets to locate these potentially unfaithful frames. We then remove spectral peaks that correspond to the extended notes in these frames. This operation reduces the mismatch caused by the sustain-pedal effects at the expense of introducing potential new mismatch caused by the removal of notes whose keys have not been released. However, we analyzed that this operation still helps the system to favor the correct score position even in this case. Experimental results on both synthetic and real piano recordings show that the proposed approach improved the alignment accuracy and robustness significantly over the baseline system.

For future work, we plan to consider other specific properties of piano music to improve the alignment performance. For example, alignment of audio and score onsets can provide “anchors” for the alignment, and we can define a special matching function that models the transient-like property to align onsets. In addition, for the sustain part, a time-varying matching function that considers the exponential energy decay would improve the alignment accuracy within a note.

6. REFERENCES

- [1] A. Arzt, G. Widmer, and S. Dixon. Automatic page turning for musicians via real-time machine listening. In *Proc. European Conference on Artificial Intelligence (ECAI)*, 2008.
- [2] J. P. Bello, L. Daudet, S. Abdallah, C. Duxbury, M. Davies, and M. Sandler. A tutorial on onset detection in music signals. *IEEE Trans. Speech and Audio Processing*, 13(5):1035–1047, 2005.
- [3] E. Benetos, A. Klapuri, and S. Dixon. Score-informed transcription for automatic piano tutoring. In *Proc. European Signal Processing Conference (EUSIPCO)*, 2012.
- [4] P. Cano, A. Loscos, and J. Bonada. Score-performance matching using HMMs. In *Proc. ICMC*, 1999.
- [5] A. Cont. Realtime audio to score alignment for polyphonic music instruments using sparse non-negative constraints and hierarchical HMMs. In *Proc. ICASSP*, 2006.
- [6] A. Cont. A coupled duration-focused architecture for real-time music-to-score alignment. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 32(6):974–987, 2010.
- [7] A. Cont, D. Schwarz, N. Schnell, and C. Raphael. Evaluation of real-time audio-to-score alignment. In *Proc. ISMIR*, 2007.
- [8] R. Dannenberg and C. Raphael. Music score alignment and computer accompaniment. *ACM Communications*, 49(8):39–43, 2006.
- [9] S. Dixon. Live tracking of musical performances using online time warping. In *Proc. International Conference on Digital Audio Effects (DAFx)*, 2005.
- [10] S. Dixon and G. Widmer. Match: A music alignment tool chest. In *Proc. ISMIR*, 2005.
- [11] Z. Duan and B. Pardo. Soundprism: An online system for score-informed source separation of music audio. *Journal of Selected Topics in Signal Processing*, 5(6):1205–1215, 2010.
- [12] Z. Duan and B. Pardo. A state space model for online polyphonic audio-score alignment. In *Proc. ICASSP*, 2011.
- [13] Z. Duan, B. Pardo, and C. Zhang. Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. *IEEE Trans. Audio Speech and Lang. Process.*, 18(8):2121–2133, 2010.
- [14] V. Emiya, R. Badeau, and B. David. Multipitch estimation of piano sounds using a new probabilistic spectral smoothness principle. *IEEE Trans. Audio, Speech, and Language Process.*, 18(6):1643–1654, 2010.
- [15] S. Ewert, M. Muller, and P. Grosche. High resolution audio synchronization using chroma onset features. In *Proc. ICASSP*, 2009.
- [16] L. Grubb and R.B. Dannenberg. A stochastic method of tracking a vocal performer. In *Proc. ICMC*, 1997.
- [17] N. Hu, R.B. Dannenberg, and G. Tzanetakis. Polyphonic audio matching and alignment for music retrieval. In *Proc. IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2003.
- [18] T. Itohara, K. Nakadai, T. Ogata, and H.G. Okuno. Improvement of audio-visual score following in robot ensemble with human guitarist. In *Proc. IEEE-RAS International Conference on Humanoid Robots*, 2012.
- [19] C. Joder, S. Essid, and G. Richard. Learning optimal features for polyphonic audio-to-score alignment. *IEEE Trans. Audio, Speech, Language Process.*, 21(10):2118–2128, 2013.
- [20] C. Joder and B. Schuller. Off-line refinement of audio-to-score alignment by observation template adaptation. In *Proc. ICASSP*, 2013.
- [21] H. M. Lehtonen, H. Penttinen, J. Rauhala, and V. Valimäki. Analysis and modeling of piano sustain-pedal effects. *Journal of the Acoustical Society of America*, 122(3):1787–1797, 2007.
- [22] B. Niedermayer, S. Bck, and G. Widmer. On the importance of real audio data for mir algorithm evaluation at the note-level - a comparative study. In *Proc. ISMIR*, 2011.
- [23] N. Orio and F. Dechelle. Score following using spectral analysis and hidden markov models. In *Proc. ICMC*, 2001.
- [24] N. Orio and D. Schwarz. Alignment of monophonic and polyphonic music to a score. In *Proc. ICMC*, 2001.
- [25] M. Puckette. Score following using the sung voice. In *Proc. ICMC*, 1995.
- [26] C. Raphael. A bayesian network for real-time musical accompaniment. In *Proc. Advances in Neural Information Processing Systems (NIPS)*, 2001.
- [27] X. Rodet and F. Jaillet. Detection and modeling of fast attack transients. In *Proc. ICMC*, 2001.
- [28] V. Thomas, C. Fremerey, D. Damm, and M. Clausen. Slave: a score-lyrics-audio-video-explorer. In *Proc. ISMIR*, 2009.
- [29] T. M. Wang, P.Y. Tsai, and A.W.Y. Su. Score-informed pitch-wise alignment using score-driven non-negative matrix factorization. In *Proc. IEEE International Conference on Audio, Language and Image Processing (ICALIP)*.

UNDERSTANDING USERS OF COMMERCIAL MUSIC SERVICES THROUGH PERSONAS: DESIGN IMPLICATIONS

Jin Ha Lee

University of Washington
jinhalee@uw.edu

Rachel Price

University of Washington
rachelpr@uw.edu

ABSTRACT

Most of the previous literature on music users' needs, habits, and interactions with music information retrieval (MIR) systems focuses on investigating user groups of particular demographics or testing the usability of specific interfaces/systems. In order to improve our understanding of how users' personalities and characteristics affect their needs and interactions with MIR systems, we conducted a qualitative user study across multiple commercial music services, utilizing interviews and think-aloud sessions. Based on the empirical user data, we have developed seven personas. These personas offer a deeper understanding of the different types of MIR system users and the relative importance of various design implications for each user type. Implications for system design include a renegotiation of our understanding of desired user engagement, especially with the habit of context-switching, designing systems for specialized uses, and addressing user concerns around privacy, transparency, and control.

1. INTRODUCTION

Designing music information retrieval (MIR) systems such as music recommenders or music management systems is challenging due to the wide variety of organizational and listening strategies of music users [3]. Although the number of studies on music users, specifically related to their needs and interactions with MIR systems, has been increasing since the early 2000s [15], our understanding on how to understand and model these users for system design is still lacking.

Previous studies of MIR system users tend to focus on investigating needs, perceptions, and opinions of general users (represented by subjects recruited online or in academic settings) or specific user groups. Studies involving specific user groups tend to investigate users based on particular demographic information or users of particular MIR systems. However, few studies attempt to categorize the "personalities" of music listeners surrounding their interaction behavior on multiple MIR systems. In addition to demographic information, what kinds of personal characteristics can we use to model commercial MIR system users for system design? Our study aims to fill this gap in prior research and answer the following questions:

RQ1. What kinds of user personas can we identify from real users of commercial MIR systems?

RQ2. What are the expressed needs and behavior of each of these user personas, and what are the implications for system design for each persona?

Our research will contribute by providing a framework for understanding users of MIR systems based on their needs and interaction behavior, beyond typical demographic information. This will help inform system designers to develop systems that are better targeted for their user groups representing particular personas, rather than creating a "one size fits all" mass production model.

2. RELEVANT WORK

2.1 HCI Studies Related to Music

A number of studies in the human computer interaction (HCI) domain explore different user behavior related to music discovery or sharing. Most of the literature focuses on testing the usability of a particular system interface, or investigating user behavior related to music discovery or sharing within a particular application.

The literature reflects a growing understanding that current music listening habits are changing. Voong & Beale [26] highlight the fact that playlist generation is done differently now than in the past, whether users create playlists by mood, theme, or other criteria. In our research, we aim to understand these criteria that are relevant to users when generating playlists and judging the playlists created by music services, and how to use those criteria to influence user experience (UX) design.

The social aspect of music consumption also seems to be a key area for investigation. Research around social playlists illustrates how friends can learn more about each other and can strengthen relationships through understanding the preferences of others ([18][21]). Bonhard et al. [4] further illustrate that "friends from whom we seek recommendations are not just a source of information for us: we know their tastes, views and they provide not only recommendations, but also justification and explanations for them. (p. 1064)" The impact of new online music repositories to people's music discovery and sharing has also been discussed in [18].

Some studies looked at the problem of how personality affects recommenders. Researchers have borrowed theories from psychology literature about personality, as in [6], exploring the impact of personality values on users' needs for recommendation diversity. Their preliminary research shows a causal relationship between personality attributes, including openness, conscientiousness, extroversion, agreeableness, and neuroticism, and users' diversity preferences when using a recommender system. In our work, we take a more empirical approach, looking at



© Jin Ha Lee, Rachel Price. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Jin Ha Lee, Rachel Price. "Understanding users of commercial music services through personas: design implications", 16th International Society for Music Information Retrieval Conference, 2015.

user data to understand various types of personas present in music services users and how the user experience can be designed to better accommodate these personas.

2.2 User Studies in MIR

Prior studies of MIR system users can be categorized into: 1) empirical investigation of music information needs, behavior, perceptions, and opinions of humans, 2) experiments, usability testing, interface design involving humans focusing on a particular MIR system, and 3) analysis of user-generated data such as queries or tags [15].

Of the first category, a few studies focus on “general music users,” often represented by queries in search engines, or human subjects recruited on various websites or a game (e.g., [8][17]). A majority of them, however, focus on a particular group of users based on demographic information. Several researchers have investigated the effects of age (e.g., young adults in [14][27]) and nationality [10][12][23]. These studies revealed that age group and cultural background do affect how people perceive, use, and search for music. A number of studies also research needs and behaviors of users in specific music-related professions (e.g., musicologists [2], DJs [20], film-makers [9]). In order to complement the findings from these studies, we look beyond demographic information and model users based on their goals/behavior within MIR systems.

A few studies focused on investigating users’ experiences with existing commercial music services, and thus are more closely related to the current paper. Barrington et al. [1] and Lamere [13] evaluated the quality of provided music recommendations or system-generated playlists. Barrington et al. [1] compare Apple iTunes’ Genius to two canonical music recommender systems: one based on artist similarity, and the other on acoustic similarity. They demonstrate the strength of collaborative filtering combined with musical cues for similarity (similar artists and other display metadata) and discuss factors that influence playlist evaluation, such as familiarity, popularity, transparency, and perceived expertise of the system. Lamere [13] also compares the playlists generated from Google’s Instant Mix, Apple iTunes, and the Echo Nest Playlist engine, and notes how personal preference of music or the context of music can affect the user experience with music services. Some factors that influence users’ evaluations of playlist (e.g., familiarity, popularity, transparency) as well as the overall perception of the quality of music service (e.g., inexpensiveness, convenience, customizability) were also identified in [1] and [17], respectively. Celma [5] discusses varied recommendation needs for four different types of listeners (i.e., savants, enthusiasts, casuals, indifferents) based on their degrees of interest in music. Lee & Price [16] also evaluated commercial music services based on Nielsen’s ten usability heuristics, advocating for more holistic evaluation of MIR systems.

Some studies focused on investigating the factors that impact people’s music listening or sharing behavior. Baur et al. [3] analyzed a sample of 310 music listening histories collected from Last.fm and 48 variables describing user and music characteristics. They found that temporal

aspects such as seasons and the degree of users’ interests in novelty were important factors affecting people’s music listening behaviors. Additionally, a number of patterns regarding users’ music seeking and consumption behavior were observed in a large-scale survey [17]: an increased consumption in mobile streaming services, an increased desire for serendipitous music discovery and music videos, as well as a strong desire to customize and personalize their music experiences.

The scope and approach of our work differ from these studies on user experience with music services in that we investigate users of ten different MIR systems (Spotify, Pandora, YouTube, Songza, SoundCloud, Grooveshark, Bandcamp, Rdio, Last.fm, iTunes), and we take a qualitative approach, asking questions and observing users’ interactions with MIR systems. Our work aims to build upon these studies and provide more detailed information about how user contexts or characteristics affect actual usage of music services.

3. RESEARCH DESIGN AND METHODS

Table 1 provides an overview of the methods and activities used for different phases for this study. The user data were collected through interviews and think-aloud sessions. All recruited participants were over 18 years old, and actively use at least one music service/application. All participants were undergraduate or graduate students at University of Washington. All the interviews were conducted between January and March 2014, either in-person or via Adobe Connect video conferencing. A total of 40 participants were interviewed and compensated with a \$15 Amazon gift card.

Methods	Activities
User interview	Semi-structured interview asking about how participants use music services and how they evaluate the quality of the services.
Think-aloud sessions	Participants narrate their actions out loud as they use their preferred music service as they would in a typical session.
Card sorting	Identify task-based user segments and create personas for each segment.

Table 1. Overview of the study design

The study session consists of two parts: first, subjects were interviewed about their preferred music services, discussing their interactions with the service, how they navigate the system, why they prefer one service over others, frustrations they experience with the service, and how they interact with the service in a typical session.

Secondly, participants were asked to “think-aloud” or narrate their actions out loud to an investigator as they use their preferred music service in a typical session. These tasks include known-item search, browsing albums, artists, or genres, interacting with recommendations, playlists, and radio stations, and other tasks as they arose. Each study session, consisting of the interview and think-aloud, lasted for approximately an hour.

The user data was used to generate a list of behaviors exhibited around MIR systems. A card sorting activity

was used to identify user groups with similar behaviors as a basis for deriving useful personas. Personas are “hypothetical archetypes of actual users (p.124)” representing their needs, behavior, and goals which allows for a goal-directed design of a system [7]. Persona development has been used to aid design and gain user insights across many fields [22], and can be beneficial for prioritizing audiences’ and users’ goals in product development [24].

We created a comprehensive list of user activities from the interview transcripts and think-aloud activities as well as the notes taken during observation. A total of 77 user behaviors related to music services were identified (e.g., read reviews, judge others’ tastes, seek recommendations). Through a card sorting activity, similar behaviors were grouped, organized, and named. We then attempted to identify which types of users would show these kinds of behaviors and tentatively named these user groups (e.g., genre fans, tech savvy). Afterwards, we identified two relevant dimensions to express the differences among these user groups organized by their common behavior, or “task-based audience segments” [28]: Companionship (willingness to engage in social aspects of music recommendation and listening: social - neutral - private) and Investment (willingness to invest time/effort to interact with the system: positive - neutral - none). As a result, we derived these seven personas:

- Active Curator: Neutral companionship + Positive investment
- Music Epicurean: Social + Positive investment
- Guided Listener: Neutral companionship + No investment
- Music Recluse: Private + Neutral investment
- Wanderer: Neutral companionship + Neutral investment
- Addict: Private + No investment
- Non-believer: Social + Neutral investment

Any user may exhibit a combination of these personas as they are not mutually exclusive. Each of these personas is explained in detail in the following section.

4. USER PERSONAS

4.1 Active Curator

This persona takes great pride in their music listening, and enjoys seeking new music and curating music he/she is already familiar with. This may come in the form of playlist creation, “saving” albums in online collections, or light music “research”, such as previewing songs or taking recommendations from friends, blogs, and live shows. Of all the personas, this one is the most actively engaged with music services (“I’m definitely an active listener 98% of the time.” (P21)).

This persona tends to utilize known-item search alongside other discovery tools, often searching rather than browsing (“I [search for song or artist] at least once a day.” (P26)). An active curator may often find discovery tools to be disappointing (“I feel like I end up listening to stuff I already know. It’s a little frustrating” (P1)). They tend to have higher expectations for music recommenda-

tion services and may not always trust a service to make good recommendations.

“One of the reasons I use these services is because I’m looking for linkages from music to music...I’m a little bit pedantic...In fact, I would love to have a little bit more information [about recommendations].” (P1)

“I would love to see the metadata that goes into choosing each song...[I’d love to] be able to pick and choose those attributes, so I could say, ‘ok, I do like those smooth jazz elements, but I don’t like the saxophone solos.’” (P30)

4.2 Music Epicurean

This persona may be considered a “music snob.” Music epicureans take an immense amount of pride in the music they collect and listen to, although they may not necessarily own all that music. Although streaming music is still an acceptable form of listening, this persona is more inclined to purchase music after listening to it than other personas as he/she genuinely cares about sound quality. A great amount of time is spent “hunting” for new music. This persona tends to focus on relationships between bands that may not be typically identified by a music recommender, such as similar “scene”, overlapping band members, and a nuanced understanding of genre relationships, and thus expresses dissatisfaction towards the given recommendations (“It looks like it’s only making recommendations of artists based on artists.” (P23)).

The music epicurean persona is unlikely to use music system recommendations; users representing this persona tend to also represent “The Non-believer” persona described below. The Music Epicurean leans on trusted sources for recommendations, whether it is a small group of friends with trusted taste or other “vetted” sources.

“I’m very self-directed in listening to music. When I listen to the radio, it’s KEXP, and it’s usually a really short amount of time in the morning. I know what I want to listen to, why am I just going to let a random radio station tell me?” (P8)

“For me it’s not really worth the time. I think it’s just going to recommend stuff that’s also tagged [similarly]...I do my own ways of [finding], and I rely on my friends and people I write with to recommend stuff...” (P6)

4.3 Guided Listener

The Guided Listener’s most prominent quality is the desire to hand over control of the music to someone else. This persona mildly enjoys radio’s serendipitous nature, may have slight preferences over genre or artist, but ultimately just wants to hear something playing. This persona is not picky; he/she may occasionally interact with a service to indicate preferences or dislikes but will not go out of his/her way to curate albums or playlists.

This persona may provide “seed” songs or artists to help a system generate a playlist or radio station, and infrequently, will browse new music or artists for fun or out of boredom. For the most part, the guided listener is a “set it and forget it” kind of person.

“It’s definitely ‘log in’, get to where I’m going, and it even goes back to the default station that I was listening to before.”

I mean, I can get this thing booted up and going within seconds, and then I'm off doing dishes or whatever, which contributes to my satisfaction. It's going to do what I want it to do immediately. Boom. Off I go." (P17)

4.4 Music Recluse

The primary characteristic of the Music Recluse is that he/she is a very private listener; this persona does not need to discuss his/her music listening habits with many people, and guards his/her privacy when using a music recommendation service. The music recluse actively avoids the social functions of music services like Spotify or Pandora and considers listening to be very personal.

This persona may have sporadic listening habits, may listen to music he/she is not proud of or would not want others to know about. Music recluses do not want people making assumptions about them based on the music they listen to.

"I would allow zero information. I already think YouTube is too invasive. They're already forcing users to create Google Plus accounts to comment on videos." (P25)

"I turned off sharing functionality. I made sure that I wasn't putting it up on Facebook or sharing it...I definitely listen to a lot of embarrassing stuff and I don't want everybody to know that. And I'm not really part of musical communities or anything, so I don't feel like scrolling through my friends' music gives me any useful information or songs to listen to." (P34)

4.5 Non-believer

The non-believer is a persona who does not believe that a machine can make adequate music recommendations for a variety of reasons: they do not understand how an algorithm can make "good" recommendations, they are able to see the limitations of recommendation algorithms, they prefer getting recommendations from friends, or they simply have not had good past experience with music recommendation services. Non-believers also have a tendency to dislike sharing personal information or listening histories with the service/system because they do not see the benefit of doing so. This persona often uses human-curated music services such as Songza or 8-Track, friends' playlists, or their own collections, which may or may not be heavily curated.

"Pandora will give me mainstream blues because it's similar rhythmically and in instrumentation, but that's not the vibe I'm looking for. It seems like they go off of something really mechanical. They're missing out on something and I don't know what it would be called, like context, and how the music makes me feel." (P23)

4.6 Wanderer

The Wanderer primarily enjoys serendipitous music discovery, and listens to new music with an open mind ("...when it recommends me things that I never would have thought of, so I think, 'yeah, I'll give it a shot'." (P11)). This persona enjoys the discovery process in general as a fun pastime, and is willing to put in some effort to discover new music. The wanderer will likely accept recommendations from a system as equally as she will accept them from a friend, a blog, or a stranger.

The wanderers tend to listen to music from a wider variety of music genres, although they may also have preferred favorites. They enjoy discovering music/artists that are less popular and are willing to listen to new artists or genres. Wanderers may like recommendations based on "playful" themes such as "Monday morning" or "Coffee music." They are more likely to use a variety of tools and also new features in the tools they regularly use.

"Honestly, the serendipity of finding new music is what I enjoy the most. Generally if I'm listening to new music it will be because a friend recommended it or I came across it on YouTube through NPR Tiny Desk or something like that. I prefer that model...I listen to pretty diverse things." (P13)

4.7 Addict

The Addict exemplifies a known-item searcher and strongly utilizes a service that features search. This persona may listen to the same song multiple times in a row, or for a whole week (e.g., "I sort of fixate." (P1)). This persona tends to use services like YouTube or Spotify where it is easy to repeat albums or songs. Their musical tastes may be all over the map, and they tend to listen to things on a whim, rather than curating any collections. They may listen sporadically, for short periods of time, and rely on easy access to music (web-based) from a variety of devices. The addict typically does not save his/her preferences by creating playlists for later access.

"I prefer Grooveshark...because I have a tendency to listen to a song, and then listen to it on repeat until I hate it forever, and Pandora doesn't let you do that at all, whereas in Grooveshark you can do that." (P23)

5. THEMES AND DESIGN IMPLICATIONS

5.1 Engagement, Ownership, and Specialization

Our user data suggest that we may need to rethink the concept of "engagement" and how that affects peoples' preferences for music services. If we consider engagement as users interacting with the system by exploring available features, then while it may be counter-intuitive, some users have no desire to engage with their preferred system. The way these users measure the success of the system is based on how little they have to interact with it.

"As soon as I figured out the basics...as soon as I found that I could look at some friends' playlists, and that I could find a few artists and make a radio station, I just, I was like, I'm done. I'm done learning how to make this work." (P1)

"There's nothing I don't like about Pandora...It might just be because I'm content enough. And I think I'm old enough, you know, I'm 45, I'm not into that 'music is my world' type of mentality. So it's not high on my list." (P17)

A strong satisficing theme was identified among these users, consistent with the finding in [16]. As long as the system does what it is "supposed to do", then it is "good enough" and users do not expect much more. This is especially exhibited by participants representing the "guided listener" persona, who tends to prefer music services like Pandora. The "addict" also tends to exhibit shallow engagement with the services. During the interview, it

became evident that most participants who can be categorized as guided listeners had never gone beyond the surface level of system. In fact, many participants discovered some of the features offered by their preferred service for the first time during the think-aloud sessions. They tend to have very specific needs and do not explore the service beyond their immediate needs.

Personas such as active curator and music epicurean showed higher levels of engagement with the systems and seemed to have a stronger sense of ownership over their music collections. Active curators in particular would spend much time curating playlists even though they do not technically “own” the music. While guided listeners would most likely be satisfied with a streaming or subscription-based model, active curators and music epicureans hesitate to abandon the collection-based model. For this reason, we expect that cloud-based music services will appeal more to the latter group of users. For them, providing a way of creating their own access points into their collection will become an important issue, as the size of their collection will continue to grow. Organizing and accessing their collection by play frequency, name of the person who recommended a track, release date, or user in households where multiple members share the music service, were some examples that respondents specifically mentioned as potentially useful.

In order to meet the needs of different personas, it may make sense to release different versions of the service/app so users can decide the appropriate version based on how much interaction they desire (“If [Spotify] had a light version then I would use that more. Like iTunes had a little mini-player, for example.” (P13)). Based on general observation, it does seem like specialization works better than generalization; each service definitely tends to attract particular types of user personas. For example, Pandora tends to attract users who do not want to spend time and effort curating music collections or listening experiences. On the other hand, Spotify users tend to invest more time in organizing their collections and providing input to improve their listening experience. Although users also rely on Spotify for music recommendations, they tend to be more critical about the results due to higher expectations. Websites like YouTube also serve a specific purpose, which is to stream videos, rather than attempting to work as some sort of Web portal that offers a variety of services. Many users, especially with need for known-item searches, will go to YouTube. Users’ strong desire to customize and personalize their music experiences was also noted in [17].

5.2 Awareness and Preserving User Trails

Another theme emerged around a user’s general awareness within a system. Most users expressed a habit of “digging” and following “wormholes” while using mid- to high-level curation tools such as Spotify, Grooveshark, and YouTube. Many of these systems do a poor job of indicating the user’s location within the site, or helping them retrace their steps, which often results in users feeling the sense of “being lost.”

“It’s constant digging. Click, click, scroll...wait, where am I? Click, scroll. For almost everything I want to do, I can never get there on the first try, or even if I get there on the first try, it feels like an accomplishment. Most of the time, I have an idea of where I am, but I don’t always know how to get back to where I was.” (P11)

“I feel like I’m not as adventurous in wormholing sometimes as I can be or want to be, because I’m afraid of getting lost. If it were a little bit easier to just go back to where you started from or some sort of chain-of-command of what you had just done that you could click through (like a breadcrumb trail), then I probably would feel a little bit more comfortable.” (P3)

This was also related to the general lack of error explanation in the systems, which would ideally help users recognize and prevent errors (“It just says, ‘There was an error.’ I almost never know what’s going on when something goes wrong.” (P11)).

Users who discussed digging, wormholes, and the like, tended to be those who actively engaged with the service. This may span across any persona, but there appears to be a correlation between concern for user trails and engaged personas like the active curator and the music epicurean. Ideally the system should support the expression and preservation of a user trail and use breadcrumb trails to give users locational clues.

Users also indicated that more transparency over recommendations would improve their likelihood of trusting the system. Not knowing why the system wants them to listen to a particular song made them less inclined to follow the recommendation, especially for the active curator, non-believer, and music epicurean personas.

“Sometimes I wonder why things are on there. I guess I need more insight on why I should choose to click on this thing...if it’s a band I’ve never heard of, I’m not going to click on it unless there’s a reason for me to...A lot of times it’s like, ‘You listened to this song by Rihanna once. All of a sudden we think you should listen to Justin Bieber.’ That doesn’t work for me.” (P31)

5.3 Privacy Concerns

Several participants discussed privacy concerns around using music services. Our data suggest that the levels of privacy concerns are possibly affected by the following three factors: a) user’s interest in/belief of a machine’s ability to accurately recommend music, b) level of understanding of privacy issues, and c) overall tech savviness. A user who has a higher interest in/belief of a machine’s ability, a better understanding of privacy issues, and is more tech savvy, tended to be more concerned about sharing their personal information. This trait was exhibited across personas regardless of music listening habits, and most dominantly in non-believers.

“When you download the software, the automatic preference is that Spotify will open every time you turn on your computer. I don’t like that. The first time I ever downloaded Spotify, that was the reason I didn’t use it [right away]. I felt like it was hijacking my computer.” (P1)

“I wouldn’t want to give a system more information about me even if it would provide a perfect playlist, because I still

want to have control of that [information]...It's creepy...I like having some degree of control and privacy." (P13)

"I'm split between 'that's really cool' and 'that's kind of creepy'. If I had the option to control it then that might be something I accept." (P30)

Being transparent about information collection and allowing more user control over privacy may help alleviate fears. This desire for control was also observed in [11], where users wanted to be in control of logging what they considered as the most private information. They found that "users prefer sharing some information automatically such as listening history, sharing some information at will and keeping some information private (p. 171)" [11]. This aligns with concerns that arose during our interviews about privacy of information or activities. While listeners may be willing to share listening history, either discretely or publicly, those same users may be concerned about other information being shared without their knowledge.

In addition to "what" is being shared, two other aspects worth noting are the different reaction to "who" is accessing users' personal information and the directionality in sharing information. There seemed to be a distinction between keeping private information from the system versus from other people. Users exhibiting the music recluse persona, for instance, were much more concerned with the latter aspect. Music epicureans seemed interested in sharing their music listening history in a limited social circle ("I talk to about five people who like the same music as me. I just feel weird about posting videos on Facebook like 'Listen to this'" (P31)). Also a number of users acted like "lurkers" in that they wanted to see what other people listen to but did not want to share their own listening habits with others.

During our work identifying the personas, we initially thought there may be a persona "Public broadcaster," someone who is very social and publicizes his or her listening choices. Careful examination of the transcripts, however, revealed that none of the users interviewed were "public broadcasters" themselves, but many made mention of that characteristic in friends or acquaintances who also use digital music services. Most of the comments alluding to the existence of this persona described how people have seen this kind of "broadcasting" behavior on social media (and were often annoyed by it). We believe that this persona may still exist, as previous research such as [11] found that their users were willing to share and seek shared information such as music listening habits, and some were already publicly doing so on websites such as Last.fm. Although users did want to keep some information private, music listening history was not such information. However, it may also be the case that we are simply seeing other's music listening history because of the default setting in some music services to publicly share such information, and as previously discussed, many users do not spend much time trying to master their service's feature settings. We plan to further explore this through a survey with a larger number of music service users.

5.4 Context-switching

In addition to the different personas, the user's context seemed critical in determining which services they use.

"It really depends. If I'm upstairs in the office and coding data, I generally listen to music that I already know and like, because I don't want it to take my focus away. If I am taking my dog for a walk or going for a drive, I may use the recommendations just to listen to new songs." (P13)

This resonates with previous MIR studies discussing how perceived qualities of music are affected by the context of the user [19], and how mood, activities, and social context among other factors influence music perception [25]. There were several aspects of user's context that seemed particularly relevant:

1) **Level of attention:** This was often dependent on other activities in which users were concurrently engaged (e.g., driving or working).

2) **Level of energy/motivation:** This is closely related to users' willingness to interact with the system. Generally, tech savvy music listeners were more willing to do so, but depending on the time of the day, this also seemed to change (e.g., acting passively while fatigued after work).

3) **Mood:** The user's mood constantly changes based on different events he/she is experiencing, and thus, the user may want to listen to songs with different "feels".

4) **Temporal aspect:** This can be seasonal or about the time of day. Depending on work schedules, the early morning or evening may be the best time for users to interact with a system. Seasonality also means that users are engaged in different activities or in seasonal moods.

User needs appear to continually shift depending on these contextual elements. A system allowing context-switching based on a combination of system logs of geo data, device usage, etc. (for attention level and temporal aspects) and users' input (for level of energy/motivation and mood) would be desirable.

6. CONCLUSION AND FUTURE WORK

In this paper, we present seven personas surrounding the use of commercial music information systems, derived from user interview data and observation of use sessions. These personas, each representing specific traits and attitudes of users, will be helpful in designing music information systems that are more highly tailored to specific user groups. Analyzing the user data made it clear that there is a relationship between persona placements on spectrums and types of services preferred. For instance, a user who is an active curator and music recluse would be more likely to use a "fringe" service such as Songza, whereas a guided listener user would likely end up relying on an online radio service like Pandora. Based on users' opinions and observations of their interactions with the services, we discussed several design implications.

In our future work, we plan to expand this study and test the applicability of these personas with a larger user population since they were derived from a relatively small sample. We will verify our results obtained from a qualitative approach by surveying a larger number of users to identify appropriate personas reflecting their characteristics, using a stratified user sample based on their most preferred commercial music service.

7. REFERENCES

- [1] L. Barrington, O. Reid, and G. Lanckriet: "Smarter than Genius? human evaluation of music recommender systems," *Proc. ISMIR*, pp. 357-362, 2009.
- [2] M. Barthelet and S. Dixon: "Ethnographic observations of musicologists at the British Library: implications for music information retrieval," *Proc. ISMIR*, pp. 353-358, 2011.
- [3] D. Baur, J. Büttgen and A. Butz: "Listening factors: a large-scale principal components analysis of long-term music listening histories," *Proc. CHI '12*, pp. 1273-1276, 2012.
- [4] P. Bonhard, C. Harries, J. McCarthy and M. A. Sasse: "Accounting for taste: using profile similarity to improve recommender systems," *Proc. CHI '06*, pp. 1057-1066, 2006.
- [5] Ò. Celma: "Music recommendation and discovery in the long tail," Ph.D. dissertation, Dept. Information & Communication Tech., UPF, 2008.
- [6] L., Chen, W. Wu and L. He: "How personality influences users' needs for recommendation diversity?" *Proc. CHI EA '13*, pp. 829-834, 2013.
- [7] A. Cooper: *The inmates are running the asylum*. Sams, Indianapolis, 1999.
- [8] D. P. W. Ellis, B. Whitman, A. Berenzweig and S. Lawrence: "The quest for ground truth in musical artist similarity," *Proc. ISMIR*, pp. 72-79, 2002.
- [9] C. Inskip, A. Macfarlane and P. Rafferty: "Music, movies and meaning: communication in filmmakers' search for pre-existing music, and the implications for music information retrieval," *Proc. ISMIR*, pp. 477-482, 2008.
- [10] X. Hu and J. H. Lee: "A cross-cultural study of music mood perception between American and Chinese listeners," *Proc. ISMIR*, pp. 535-540, 2012.
- [11] T. Kärkkäinen, T. Vaittinen and K. Väänänen-Vainio-Mattila: "I don't mind being logged, but want to remain in control: a field study of mobile activity and context logging," *Proc. CHI '10*, pp. 163-172, 2010.
- [12] K. Kosta, Y. Song, G. Fazekas and M. B. Sandler: "A study of cultural dependence of perceived mood in Greek music," *Proc. ISMIR*, pp. 317-322, 2013.
- [13] P. Lamere. How good is Google's Instant Mix? <http://musicmachinery.com/2011/05/14/how-good-is-googles-instant-mix/>
- [14] A. Laplante and J. S. Downie: "The utilitarian and hedonic outcomes of music information seeking in everyday life," *Library and Information Science Research*, Vol. 33, No. 3, pp. 202-210, 2011.
- [15] J. H. Lee and S. J. Cunningham: "Toward an understanding of the history and impact of user studies in music information retrieval," *Journal of Intelligent Information Systems*, Vol. 41, pp. 499-521, 2013.
- [16] J. H. Lee and R. Price: "User experience with commercial music services: an empirical exploration," *JASIST*, DOI: 10.1002/asi.23433, 2015.
- [17] J. H. Lee and N. M. Waterman: "Understanding user requirements for music information services," *Proc. ISMIR*, pp. 253-258, 2012.
- [18] T. W. Leong and P. C. Wright: "Revisiting social practices surrounding music," *Proc. CHI '13*, pp. 951-960, 2013.
- [19] M. Lesaffre, B. De Baets, H. De Meyer and J-P. Martens: "How potential users of music search and retrieval systems describe the semantic quality of music," *JASIST*, Vol. 59, No. 5, pp. 695-707, 2008.
- [20] J. Lingel: "When organization becomes an extension of your brain: DJs, music libraries and information practices," *Proc. iConference*, pp. 366-376, 2013.
- [21] K. Liu and R. A. Reimer: "Social playlist: enabling touch points and enriching ongoing relationships through collaborative mobile music listening," *Proc. MobileHCI 2008*, pp. 403-406, 2008.
- [22] T. Matthews, T. Judge and S. Whittaker: "How do designers and user experience professionals actually perceive and use personas?" *Proc. CHI '12*, pp. 1219-1228, 2012.
- [23] E. Nettamo, M. Nirhamo and J. Häkkinen: "A cross-cultural study of mobile music: retrieval, management and consumption," *Proc. OZCHI '06*, pp. 87-94, 2006.
- [24] L. Nielsen and K. S. Hansen: "Personas is applicable: a study on the use of personas in Denmark," *Proc. CHI '14*, pp. 1665-1674, 2014.
- [25] M. Schedl: "Ameliorating music recommendation," *Proc. MoMM '13*, pp. 3-9, 2013.
- [26] M. Voong and R. Beale: "Music organisation using colour synaesthesia," *Proc. CHI EA '07*, pp. 1869-1874, 2007
- [27] J. Woelfer and J. H. Lee: "The role of music in the lives of homeless young people: a preliminary report," *Proc. ISMIR*, pp. 367-372, 2012.
- [28] I. Young: *Mental models: aligning design strategy with human behavior*. Rosenfeld Media, 2007.

CORPUS-BASED RHYTHMIC PATTERN ANALYSIS OF RAGTIME SYNCOPATION

Hendrik Vincent Koops
Utrecht University
h.v.koops@uu.nl

Anja Volk
Utrecht University
a.volk@uu.nl

W. Bas de Haas
Utrecht University
w.b.dehaas@uu.nl

ABSTRACT

This paper presents a corpus-based study on rhythmic patterns in the RAG-collection of approximately 11.000 symbolically encoded ragtime pieces. While characteristic musical features that define ragtime as a genre have been debated since its inception, musicologists argue that specific syncopation patterns are most typical for this genre. Therefore, we investigate the use of syncopation patterns in the RAG-collection from its beginnings until the present time in this paper. Using computational methods, this paper provides an overview on the use of rhythmical patterns of the ragtime genre, thereby offering valuable new insights that complement musicological hypotheses about this genre. Specifically, we measure the amount of syncopation for each bar using Longuet-Higgins and Lee’s model of syncopation, determine the most frequent rhythmic patterns, and discuss the role of a specific short-long-short syncopation pattern that musicologists argue is characteristic for ragtime. A comparison between the ragtime (pre-1920) and modern (post-1920) era shows that the two eras differ in syncopation pattern use. Onset density and amount of syncopation increase after 1920. Moreover, our study confirms the musicological hypothesis on the important role of the short-long-short syncopation pattern in ragtime. These findings are pivotal in developing ragtime genre-specific features.

1. INTRODUCTION

This paper presents a corpus-based study into rhythmic patterns in a ragtime corpus (RAG-collection) of approximately 11000 pieces (rags), collected by an international group of ragtime lovers¹. The RAG-collection (RAG-C) was introduced in [16], together with an overview of open questions and a computational confirmation of musicological hypotheses of ragtime music.

Esparza et al. [3] argue that in MIR, genre classification has often been used as a proxy for measuring the success

¹Ragtime Admirers Group, see <http://ragtimecompendium.tripod.com/>



© Hendrik Vincent Koops, Anja Volk, W. Bas de Haas. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Hendrik Vincent Koops, Anja Volk, W. Bas de Haas. “Corpus-Based Rhythmic Pattern Analysis of Ragtime Syncopation”, 16th International Society for Music Information Retrieval Conference, 2015.

of rhythmic similarity measures, based on the assumption that “*rhythmic content is more or less homogeneous within certain musical styles*”. Their research shows that even for dance music this is often a problematic assumption. Therefore, a better understanding of the relation between rhythm and genre is important. Musicologists and ragtime fans have argued that rhythmic patterns and syncopation provide the most distinct features of the genre [1]. Edward Berlin argues that syncopation is “*at the core of the contemporary understanding of ragtime*” [2]. However, musicologists also argue that the use of rhythmic patterns has not been stable within the development of the genre over time. Therefore, we investigate ragtime’s *syncopation*, its typical rhythmical *patterns* and their evolution over time.

Huron et al. [7] have shown for related genres that syncopation increases through history, something we hypothesize will be the case for ragtime as well. We reflect on the rhythmical *patterns* of the genre: what are the most characteristic rhythmic patterns used in ragtime syncopation, and does their use change over time. Berlin [2] argued for the importance of a specific short-long-short pattern in the ragtime genre, of which Volk and De Haas [16] showed that its use increased through history. We extend the research in [16] by investigating *all* patterns, to find the relative importance of this specific pattern. We hypothesize that compared to other patterns appearing in ragtime syncopation, this short-long-short pattern is one of the most characteristic patterns for the ragtime genre.

Our corpus-based study of syncopation complements extensive research on syncopation in music cognition, in which predominantly short rhythmic patterns are studied. Syncopation is considered to create violations in listeners’ expectations [11], to contribute to rhythmic complexity [17] and to contribute to a sense of groove in music [12, 13]. Studying syncopation for entire compositions instead of short stimuli contributes to understanding how much violation and complexity is used in real compositions of a genre that is considered to be “highly syncopated”.

Contribution. The contribution of this paper is threefold. We present a first full, systematic analysis of all rhythmic patterns in melodies appearing in a large corpus of ragtime music. Through a statistical analysis of the frequency of patterns over time, this study shows which patterns are more important in different time periods. Second, by using a formal model of syncopation, this study is able to focus on the syncopated parts of rags, commonly thought to be the most characteristic element of ragtime

music. Through this model, it shows the increase of syncopation use together with its most important rhythmical patterns over time. Third, a tactus finding algorithm is introduced that is capable of correctly identifying the number of beats in a bar of a ragtime piece. These three contributions are pivotal in understanding the characteristic features of ragtime music.

Synopsis. The remainder of this paper is structured as follows: Section 2 provides an introduction to ragtime music and its use of syncopation. Section 3 details the main methodology for analysing patterns and syncopation in the RAG-C. Section 4 details the results of syncopation analysis and pattern finding. The paper closes with conclusions and discussion in section 5.

2. RAGTIME

Musicologists agree that ragtime's most striking element can be found in its use of syncopated rhythmical patterns. Berlin [2] even argues that other musical features are of hardly any importance: ragtime music has no unique musical form, and its melodies do not bear any distinctive traits (except with regard to rhythm). Although rags with hardly any syncopation exist, musicologists do agree that syncopation is the dominant and distinctive element in the evolution of the ragtime genre. It is therefore that a study into ragtime will invariably involve the analysis of rhythmical patterns and syncopation.

In this research, we divide the history of ragtime music into two eras: the pre-1920 *ragtime era* and the post-1920 *modern era*. The two eras are distinguished by a remarkable increase in rhythmic experimentation and syncopation around 1920 [8, page xix]. This change was in part influenced by the French Impressionist music and piano performers mimicking the very complex rhythms of piano-roll music that were in style.

2.1 Syncopation

Syncopation is “the displacement of the normal musical accent from a strong beat to a weak one”, often used by composers to avoid regular rhythm by varying position of the stress on notes [14]. Musicologists have argued that ragtime's main identifying trait is its “ragged”, or *syncopated* rhythm. A specific syncopated pattern is thought to be of extra importance by Harer [6] and Berlin [2]: the ‘short-long-short’ 121 pattern. The 121 pattern appears as

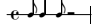
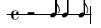
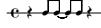

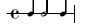



1. Untied (in *U* bar parts):
 | IOIOOOIO OOOOOOOO |
 | OOOOOOOO IOIOOOIO |
2. Tied (in *T* bar parts):
 | OOOOIOIO OOIIOOOO |
 | OOOOIOIO OOIIOOOO |
3. Augmented:
 | IOOOIOOO OOOOIOOO |



Figure 1: 121 patterns in musical notation (left) and equivalent binary onset pattern (right).

 in $\frac{4}{4}$ and as  in $\frac{3}{4}$. The next sections detail three variants of the 121 pattern: untied, tied and augmented. Examples of these three types of syncopation in $\frac{3}{4}$ can also be found in Figure 1.

Untied syncopation. In *untied* syncopation, a pattern starts on a strong metrical position and does not pass over a bar line. In $\frac{3}{4}$, the pattern either starts on the first quarter note position or the second quarter note position. In $\frac{4}{4}$, the 121 pattern () would start on either the first quarter note position or the third quarter note position. This is visualized in Figure 2 as the *U* bar parts. This way, the 121 pattern always constitutes the first or second half of a bar. Musicologists have argued that this type of syncopation is more characteristic of rags from the early pre-1920 ragtime era, being more prominent at the turn of the century [10], [2, p. 84].

Tied syncopation. *Tied* syncopation refers to a pattern starting on a weak metrical position. Just like untied syncopation, the tied version appears in two variants: either creating a tied note over the center of the bar, or over the barline to the next bar. This is visualized in Figure 2 as the *T* bar parts. In $\frac{4}{4}$ this means the pattern starts at the second or fourth quarter note position. In $\frac{3}{4}$ this means the pattern starts at the first or third eighth note position.

The *tied* pattern was found to increase during the pre-1920 era by [16]. Musicologists have argued that composers increasingly relied on tied syncopation in the late 1910s and 1920s as the ragtime style matured [10, p. 76].

Augmented syncopation. A third version of syncopation often found in ragtime music is called *augmented* syncopation. This type of syncopation augments the 121 to the length of a complete bar (3 of Figure 1). The augmented pattern appears as  in $\frac{4}{4}$, and as  in $\frac{3}{4}$. This results in a weaker syncopated pattern, which is more characteristic of early ragtime era [2, page 83], but became relatively rare after 1903.

3. METHODOLOGY

This study investigates the use of syncopation and rhythmical patterns in the RAG-C, and how these change over time. We hypothesize that syncopation is an important feature of the ragtime genre, that increases over time. To test this hypothesis, we first extract rhythmical onset patterns rags in the RAG-C, as detailed in Section 3.1. Then, to be able to group the onsets in *bars* for analysis, the number of beats per bar need to be determined. To achieve this, a tactus finding algorithm (Section 3.2) that finds the number of beats in a bar of a ragtime piece is implemented.

Differentiating between bars with and without syncopation provides insight in the patterns that are most important within ragtime syncopation. To measure the degree of syncopation of a bar, a model by Longuet-Higgins and Lee is used, as detailed in Section 3.3. These syncopation measurements are then used in a pattern recognition step (Section 3.4), to find the frequencies of all possible patterns and the relative 121 frequencies. The following sections describe each of these steps in detail.

3.1 Onset Extraction

Characteristic of ragtime music is a ragged or syncopated melody over a stable accompaniment that reinforces the meter. The importance of first separating a piece into its individual rhythmic layers for syncopation measurements was shown in [13]. Therefore, to be able to analyse syncopation of the melody of a rag, we split it from its accompaniment. The accompaniment is used in a tactus finding step (detailed in Section 3.2), and the melody is used in a pattern finding step (section 3.3).

The melody and accompaniment are split using the skyline with dip-detection method detailed in [15, 16], which performs a near-perfect splitting of a melody and its accompaniment on a subset of the RAG-C. To be able to analyse rhythmical patterns properly, both the melody and accompaniment are quantized. We use the technique described by Volk & De Haas [16], with the exception of using four bins per quarter note, instead of twelve. This results in quantisation to a sixteenth note grid, which we can apply to the formal model of syncopation described in Section 3.3. Because of different *normalized average quantisation deviations* (the average deviation of notes divided by the MIDI quarter note length, see [16]) in different files in the dataset, we keep track of the quantisation error, and disregard all MIDI files with a normalized average quantisation error above 2%.

This results in two sequences of onsets per rag, one representing the rhythm of the melody, and one representing the rhythm of the accompaniment. The onsets in the sequences are represented with \uparrow 's as sounding events and \circ 's as non-sounding events. See the bottom two rows of the tree in Figure 2 for an example with its music notation equivalent.

3.2 Tactus Finding

This study analyses the onset patterns that appear in syncopated *bars* of rags. The method in Section 3.1 results in a sequence of onsets, therefore we need a way to segment this sequence into bars. One way to achieve this is to use the annotated MIDI time signature of the rags, but from a manual inspection this information was found to be not always reliable. Therefore, a tactus finding algorithm is created that is able to find the number of beats in a bar. This information is used to group the right number of onsets into bar representations: from a sequence of onsets to segments representing bars.

Two features of ragtime music facilitate time signature detection from onset patterns with greater ease, compared to other genres. First, most rags are written in either $\frac{2}{4}$ or $\frac{4}{4}$, other meters are rare. Secondly, a characteristic feature of ragtime is a stable metre pattern in the accompaniment underneath a syncopated melody [9]. As a general rule, the accompaniment “reinforces the meter with a regular alternation of low bass notes or octaves on the beat, alternating with mid-range chords between the beats” [1].

In $\frac{4}{4}$ (and $\frac{3}{2}$), this alternation appears as $\downarrow\downarrow\downarrow\downarrow$. In $\frac{2}{4}$, this pattern appears as $\downarrow\downarrow\downarrow$. This pattern can be used to estimate the number of beats in a bar for duple time signatures.

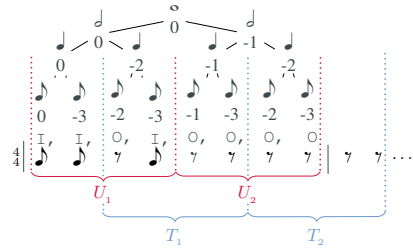


Figure 2: Example of a hierarchical metric tree with values. U and T denote the bar parts where we search for patterns. The onset pattern (\uparrow 's and \circ 's) represents the 121-pattern ($\downarrow\downarrow\downarrow\downarrow$) in a U part of a bar. The LHL value of this bar is $((-2) - (-3)) + ((-1) - (-3)) = 1 + 2 = 3$.

The algorithm presented in this paper finds the number of beats of a rag by assuming that in $\frac{2}{4}$ the onset density in the accompaniment is higher than in $\frac{4}{4}$. The onset density d for a sequence of onsets is calculated by dividing the number of onsets on even positions by the number of onsets on odd positions. If d is larger than a certain threshold, it is assumed that the onset density is low, and a time signature of $\frac{4}{4}$ is assumed. If the fraction is lower than a threshold, the onset density is high and the time signature is assumed to be $\frac{2}{4}$. Using this information, the onset sequence is either segmented in 16 onsets per bar in the case of $\frac{4}{4}$ and 8 onsets per bar in the case of $\frac{2}{4}$. These bar onset patterns are then used in a next step in which the amount of syncopation is measured, as explained in Section 3.3.

Evaluation. The tactus finding algorithm is evaluated in an experiment using 200 randomly selected rags from the RAG-C. After quantization and selecting rags with a normalized average quantization error below 2%, 72 rags remain. The rags are manually annotated with their correct time signature by a music expert. Using the technique described in 3.2, the algorithm predicts the correct number of beats in a bar in 92% (66) of the rags using a threshold of $d = 0.8$. Of the six songs that were incorrectly identified, two are not in a $\frac{2}{4}$ or $\frac{4}{4}$ time signature ($\frac{3}{2}$ and $\frac{6}{8}$) and four lack the typical accompaniment pattern. These results show that this method is highly useful as a preprocessing step for segmenting onsets into bars.

3.3 Longuet-Higgins & Lee Syncopation Measurement

For pattern analysis, we differentiate between bars with and without syncopation and analyse the former, to find its most characteristic patterns. A formal model of syncopation introduced by Longuet-Higgins & Lee (LHL) [11] provides a numerical representation of syncopation in a bar by assuming that a rhythm in a meter is interpreted by a listener by minimizing the amount of syncopation. In an experimental comparison between different syncopation measurements, Gómez et al. [5] found that the LHL agrees closely with the human judgement of syncopation. The notion of minimizing syncopation is expressed in the

algorithm, in which syncopation is defined to occur when a note occurs on a weaker position than its succeeding rest (or tied note). This was also shown empirically by Fitch et al. [4], who showed that the recall of a rhythm decreased with higher LHL syncopation.

The LHL model computes syncopation using a tree of metric hierarchy (see Figure 2 for an example). This tree is built to a minimal depth needed to represent the notes. For example, if a $\frac{1}{4}$ bar only contains two half notes ($d\bar{d}$), a tree of depth 1 is used. In case a note appears on a deeper level, a deeper tree is used (e.g. depth 4 in $d\bar{7}d\bar{d}$).

The nodes of the tree are populated with values k given to the left children and $-d$ to right children, where k is the value of the parent of a node and $-d$ is the negative value of the depth of the tree at that node. The value of the root of the tree is 0.

In the LHL model, syncopation occurs where a note (I) with a lower value is followed by a rest (O) with a higher value. The example in Figure 2 contains two of these (I, O) pairs, the second eighth note followed by a rest, and the third eighth note followed by a rest. The amount of syncopation for a pair is the difference in values: $\text{O}-\text{I}$, $(-2) - (-3) = 1$ for the first example. The total syncopation value of an entire bar is the sum of syncopation pairs within that bar:

$$\sum_{i=1}^n (\nu(\text{O}_{i+1}) - \nu(\text{I}_i)) \quad \text{if } \nu(\text{I}_i) < \nu(\text{O}_{i+1}) \quad (1)$$

where the subscript denotes the i^{th} position in the bar of length n and $\nu(\varphi)$ denotes the metric tree value of φ .

3.4 Pattern finding

To find the frequencies of onset patterns in the RAG-C, a pattern finding algorithm is created. We are interested in the bar parts where the tied, untied and augmented 121 pattern can appear. Therefore, this algorithm finds the frequency of candidate patterns in U and T bar parts (see Figure 2). With this quantitative measurement of pattern frequencies, we measure whether the 121 pattern is indeed characteristic for ragtime music in these bar parts, and what other patterns are important. To be able to search for patterns in U bar parts, each bar from the RAG-C is concatenated with half of the bar that follows it.

To find the frequencies of patterns in U and T , all possible combination of I 's and O 's are generated for the length of half a bar. For example, in the case of a bar in $\frac{1}{4}$ quantized on sixteenth notes, a full bar contains 16 onsets. Therefore, all candidate patterns (ρ) of length 8 are generated: $[\text{O}, \text{O}, \text{O}, \text{O}, \text{O}, \text{O}, \text{O}, \text{O}] \dots [\text{I}, \text{I}, \text{I}, \text{I}, \text{I}, \text{I}, \text{I}, \text{I}]$. The frequency of each candidate pattern $\rho \in \rho$ is calculated by counting how often each one appears in one of the U and T parts, normalized over the total number of bars. Calculating the frequency of all pattern results in distributions of patterns in U and T bar parts.

4. RESULTS

This section describes statistics of syncopation and the results of finding the most frequently used patterns in U and

T parts of syncopated bars. First, results of finding the most frequent patterns in the entire RAG-C are presented (Section 4.1). Then, the RAG-C is split into rags from the ragtime era (before 1920) and the modern era (after 1920) to find which patterns are characteristic for these eras. These results are presented in Section 4.2. In the next sections, \bar{x} denotes an average and σ denotes a variance.

4.1 Syncopation in the RAG-C

From the RAG-C, 356519 bars are extracted, of which 46% (163197) are syncopated (i.e. $\text{LHL} > 0$). The average LHL value of syncopated bars is $\bar{x} = 2.02$, $\sigma = 1.08$. The largest LHL syncopation is 15, corresponding with only 28 bars in the RAG-C. Nevertheless, a little over half of the bars (54% = 193322) in the RAG-C is devoid of any syncopation (i.e. $\text{LHL}=0$).

Finding the most frequently used patterns in T and U bar parts of bars with $\text{LHL} > 0$ yields the results in Figure 3. Note that patterns are part of a syncopated bar, and not necessarily syncopated themselves. For example, a bar consisting of $|\text{I}\text{O}\text{I}\text{O}\text{O}\text{O}\text{I}\text{O} \text{O}\text{O}\text{O}\text{O}\text{O}\text{O}\text{O}|$ is syncopated because of the 121 pattern in the first half of the bar ($\text{I}\text{O}\text{I}\text{O}\text{O}\text{O}\text{I}\text{O}$), however, the second half ($\text{O}\text{O}\text{O}\text{O}\text{O}\text{O}\text{O}$) is devoid of any syncopation.

The figure shows that the 121 pattern appears as the most frequent pattern in T bar parts, and as a third most frequent pattern in U bar parts. This affirms the hypothesis that when taking rags from all time periods in consideration, the 121 pattern is indeed one of the most characteristic ragtime patterns. The figures show that for the whole RAG-C, the 121 is more characteristic in T than in U .

Ragtime and Modern era. We split the RAG-C into pre-1920 *ragtime era* bars and post-1920 *modern era* bars,

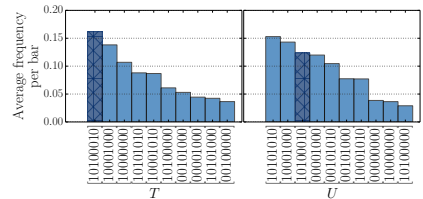


Figure 3: The 10 most frequent patterns in T and U parts of bars with $\text{LHL} > 0$. The 121 pattern is visualized darker.

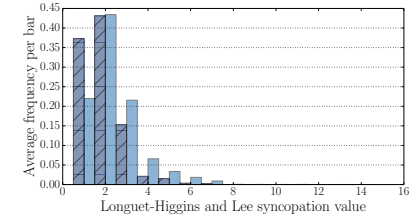


Figure 4: Percentage of bars with $\text{LHL} > 0$ in ragtime era (dark) and modern era (light). Values > 7 are too small to be visible.

to find the change in syncopation degree over time. The average LHL syncopation of a ragtime era bar with $LHL > 0$ is $\bar{x} = 1.9$, and $\bar{x} = 2.4$ in the modern era. In a Wilcoxon test for the null hypothesis that two related paired samples come from the same distribution, we find that $p \ll 0.001$, which shows that the modern era is significantly stronger syncopated. Also taking into account the non-syncopated bars shows a significant difference, with $\bar{x} = 0.83$ (ragtime era) and $\bar{x} = 1.26$ (modern era), again with $p \ll 0.001$.

Figure 4 shows distribution of LHL syncopation found in syncopated bars from these two eras. The figure shows that bars with $LHL=1$ are more common in the ragtime era, and $LHL=2$ is almost equally common in the ragtime era as in the modern era. Nevertheless, it also shows that bars with $LHL > 3$ are more characteristic for bars from the modern era. Bars with $LHL > 5$ occur twice as often in the modern era compared to the ragtime era.

Syncopation per rag. To find the distribution and degree of syncopated bars of complete rags in the RAG-C, we computed statistics on rags. The average syncopation per rag for the whole RAG-C is $\bar{x} = 0.95$, $\sigma = 0.6$. An LHL value of 1 roughly corresponds with a single syncopation inside one of the U parts, resulting in a bar of ♪♩ .

For the ragtime era, the average syncopation per rag is $\bar{x} = 0.85$, $\sigma = 0.52$. For the modern era this is $\bar{x} = 1.28$, $\sigma = 0.74$. Therefore, in the modern era, syncopation more often appears on weaker metric positions that correspond with lower values in the LHL tree, thereby increasing the LHL value of the bar. An example of this is ♪♩♩♪ . For both eras, we find that the number of syncopated bars per rag is around 50%, which means that not the number of syncopated bars increases, but the use of syncopation inside bars does. We found that the difference in syncopation between ragtime and modern era to be highly significant with $p \ll 0.001$. When only taking into account the syncopated bars, we find $\bar{x} = 1.84$, $\sigma = 0.54$ per rag for the ragtime era, and $\bar{x} = 2.29$, $\sigma = 0.67$ per rag for the modern era, again with $p \ll 0.001$.

Both the statistics on rags and bars show that overall, stronger syncopation is more characteristic of modern era rags. In the modern era, syncopation occurred more often on weaker metric positions, thereby increasing the LHL syncopation. The next section details the difference in patterns found between these eras.

4.2 Frequent Patterns in Ragtime and Modern Era

To find a change in pattern use in syncopation over time, we look at the patterns found in syncopated bars from the ragtime era and modern era. Figure 5 and Figure 6 show the 10 most frequent patterns found in U and T bar parts in the figures, the 121 pattern is visualized darker.

Patterns appearing in U bar parts. The left side of Figure 5 shows that the 121 pattern in U occurs more frequently in the modern era compared to the ragtime era. Secondly, it shows that the 121 pattern in U also became more important over time compared to other patterns. Although the 121 pattern in the modern era is the second most frequent pattern, the difference between the first and third

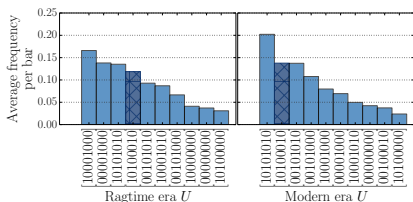


Figure 5: The 10 most frequent U patterns found in bars with $LHL > 0$ in ragtime and modern era. 121 pattern is visualized darker.

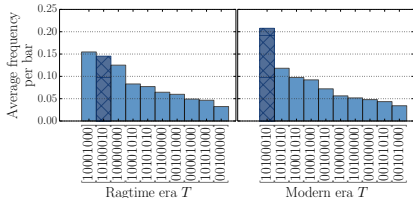


Figure 6: The 10 most frequent T patterns found in bars with $LHL > 0$ in ragtime and modern era. 121 pattern is visualized darker.

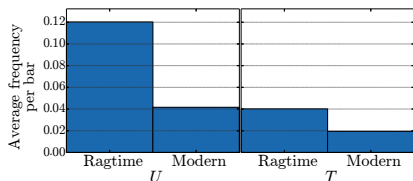


Figure 7: Frequency of augmented pattern in U (left) and T (right) bar parts of ragtime era and modern era bars.

most frequent pattern (♪♩) is minimal. The figure affirms the hypothesis that the 121 pattern is an important pattern amongst other patterns, and its importance for the ragtime genre in U increased over time.

Another difference between the ragtime and modern era can be found in the onset density and metrical position of onsets. In the ragtime era, the top most frequent patterns have a lower onset density and have more notes on strong metrical position, indicating that the patterns used in the ragtime era are less complex. In the modern era, the top most frequent patterns are more dense and have more notes occurring on weaker metrical positions.

Patterns appearing in T bar parts. Figure 6 shows the 10 most frequent patterns found in T parts of syncopated bars. The figure shows that the 121 pattern is an important pattern in T , being the second most frequent pattern in the ragtime era and by far the most frequent pattern in the modern era. These results affirm the hypothesis that both the importance and use of the 121 pattern in T has increased over time. In a study by Huron et al. [7], a top 10 of most frequently found syncopation patterns in sound recordings of American popular music spanning the period

1890 to 1939 is presented. The most frequently found syncopation pattern appears here as well, as the fifth most important pattern in the modern era bars: $7\downarrow$ (OOIOOOIO).

Furthermore, an increase in onsets on weak metrical positions is observed. The first couple of most frequent patterns in the ragtime era are simple rhythms on strong metrical positions. Conversely, in the ragtime era we observe denser patterns. Both the increase of 121 use and use of denser patterns is in line with the argument of Jasen [8] that after around 1920, rags became more difficult to play, because “[...] writers were no longer writing for the at-home amateur pianist, [...], but were writing for themselves and for other professional performers”.

U and T patterns between eras. A comparison between the leftmost figures of Figure 5 and Figure 6 shows that for both T an U , the most frequent ragtime era pattern is the same regular sparse pattern (\downarrow). Since this pattern itself is not syncopated according to the LHL model, this shows that in syncopated bars from the ragtime era most often only one half of a bar contains syncopation, indicating a lower amount of overall syncopation compared to the modern era. The 121 pattern is an important pattern for the ragtime era, being the fourth most important pattern in U bar parts and second most important pattern in T bar parts. The use of the 121 pattern increased over time, both in U as in T . In the modern era, the 121 pattern is by far the most important pattern in T bar parts, and the second in U .

Overall, the most frequent patterns in the ragtime era show more sparse onset patterns on strong metrical positions, indicating simpler rhythms. The most frequently observed pattern in the ragtime era (\downarrow) corresponds with a part of the third most common syncopation pattern found by Huron et al. [7] in sampled music from 1890 to 1939. Nevertheless, the research by Huron et al. does not focus specifically on ragtime music, so further cross-genre research is needed to find if this pattern is specifically important for ragtime. Conversely, it is observed that the patterns in the modern era are more dense. Onsets appear more frequently on weaker metrical positions, increasing the complexity of patterns in terms of onset density over time. This agrees with the musicological hypotheses that earlier ragtime is simpler, and the exceptional renewed rhythmical creativity from around 1920 onwards [2, 8].

Augmented syncopation. Figure 7 shows the frequencies of the augmented 121 pattern in U and T bar parts. In U , a difference of around 60% is observed, which reflects the argument by Berlin [2] that the pattern becomes “quite rare” at the end of the ragtime era. Although rare to begin with in T , the occurrence drops with 50% in the modern era compared with the ragtime era. Care should be taken with drawing conclusions from these results because of the low frequency. The observations on the augmented pattern underline the overall trend of ragtime moving towards using onsets on weaker metrical positions and increased onset density of patterns, thereby becoming more syncopated.

5. DISCUSSION AND CONCLUSION

Through this study, we were able to confirm new and existing hypotheses on increasing syncopation and rhythmic pattern use in the ragtime genre.

Ragtime music is often described as ‘highly syncopated’. Through the RAG-C, we showed for the first time that in a large corpus this translates into about half of the bars of rags being syncopated. Musicologists have argued that syncopation is important for the ragtime genre. Through the computational means in this study, we can affirm the hypothesis syncopation is a characteristic feature of the genre. We can also confirm the hypothesis that the amount of syncopation is not stable over time, but increased after 1920. More specifically, by exploring this notion of increased syncopation we discovered that the number of syncopated bars is approximately equal in ragtime and modern era rags, but that the LHL values of bars increases.

In an analysis of all patterns used in ragtime syncopation, we showed the top 10 most frequently used patterns in syncopated bars. We found that over time, onset patterns became more dense with more notes on weaker metrical positions. This finding is consistent with the increase of LHL. We can affirm the findings by Volk and De Haas [16] on the increase of 121 after the ragtime era. In addition, we showed that the 121 pattern is a highly important rhythmic pattern for the genre, being one of the most frequently used patterns compared to all other patterns.

Our corpus-based study on syncopation complements studies in music cognition research, which have investigated syncopation’s role on violating listeners’ expectations, thereby contributing to listening pleasure of the music [17]. These studies are predominantly carried out on short rhythmic stimuli. Understanding the full power of syncopation requires its study within entire compositions as realized within this paper. Violating listeners’ expectation through the use of syncopation in this ragtime corpus is realized on average in half of the bars in the melody. Whether or not there are other genres that use even more violations, while still providing a clear sense of meter, will have to be addressed in future research.

To study what ‘highly syncopated’ means in the context of other genres, we plan on comparing the amount of syncopation in the RAG-C to other genre datasets. Furthermore, a study into the use of the 121 pattern in other genres would shed light on the relative importance of the pattern to ragtime and other genres.

6. ACKNOWLEDGEMENTS

We thank R.C. Veltkamp and D. Bountouridis for providing valuable comments on an earlier draft on this text. The authors would like to thank anonymous reviewers for their valuable comments and suggestions. H.V. Koops, A. Volk and W.B. de Haas are supported by the Netherlands Organization for Scientific Research, through the NWO-VIDI-grant 276-35-001 to A. Volk.

7. REFERENCES

- [1] Edward A. Berlin. *Ragtime*. Oxford Music Online. Grove Music Online. Oxford University Press. Addressed: April 24, 2015.
- [2] Edward A. Berlin. *Ragtime: A musical and cultural history*. Univ of California Press, 1984.
- [3] Tlacacl Miguel Esparza, Juan Pablo Bello, and Eric J. Humphrey. From genre classification to rhythm similarity: Computational and musicological insights. *Journal of New Music Research*, 44(1):39–57, 2015.
- [4] W. Tecumseh Fitch and Andrew J. Rosenfeld. Perception and production of syncopated rhythms. *Music Perception: An Interdisciplinary Journal*, 25(1):pp. 43–58, 2007.
- [5] Francisco Gómez, Eric Thul, and Godfried T Tousseint. An experimental comparison of formal measures of rhythmic syncopation. In *Proceedings of the International Computer Music Conference*, pages 101–104, 2007.
- [6] Ingeborg Harer. *Ragtime: Versuch einer Typologie*. Schneider, 1989.
- [7] David Huron and Ann Ommen. An empirical study of syncopation in american popular music, 1890–1939. *Music Theory Spectrum*, 28(2):211–231, 2006.
- [8] David A. Jasen. *Ragtime: an encyclopedia, discography, and sheetography*. Taylor & Francis, 2007.
- [9] Samuel A. Floyd Jr. and Marsha J. Reisser. The sources and resources of classic ragtime music. *Black Music Research Journal*, 4:pp. 22–59, 1984.
- [10] Stanley V. Kleppinger. On the influence of jazz rhythm in the music of Aaron Copland. *American Music*, 21(1):pp. 74–111, 2003.
- [11] H. Christopher Longuet-Higgins and Christopher S. Lee. The perception of musical rhythms. *Perception*, 11(2):115–128, 1982.
- [12] Guy Madison and George Sioros. What musicians do to induce the sensation of groove in simple and complex melodies, and how listeners perceive it. *Frontiers in Psychology*, 5(894), 2014.
- [13] George Sioros, André Holzapfel, and Carlos Guedes. On measuring syncopation to drive an interactive music system. In *Proceedings of the 13th International Society for Music Information Retrieval Conference, ISMIR 2012, Porto, Portugal, October 8–12*, pages 283–288, 2012.
- [14] Syncopation. *The Oxford Dictionary of Music, 2nd ed. rev.* Oxford University Press, 2006.
- [15] Alexandra L. Uitenbogerd and Justin Zobel. Manipulation of music for melody matching. In *Proceedings of the sixth ACM international conference on Multimedia*, pages 235–240. ACM, 1998.
- [16] Anja Volk and W. Bas de Haas. A corpus-based study on ragtime syncopation. In *Proceedings of the 14th International Society for Music Information Retrieval Conference, ISMIR 2013, Curitiba, Brazil, November 4–8, 2013*, pages 163–168, 2013.
- [17] Maria A.G. Witek, Eric F. Clarke, Mikkel Wallentin, Morten L. Kringselbach, and Peter Vuust. Syncopation, body-movement and pleasure in groove music. *PLoS one*, 9(4):e94446, 2014.

Oral Session 3
Melody & Voice

COMPARING VOICE AND STREAM SEGMENTATION ALGORITHMS

Nicolas Guiomard-Kagan
MIS, U. Picardie Jules Verne
Amiens, France

Mathieu Giraud
CRIStAL (CNRS, U. Lille)
Lille, France

Richard Groult
MIS, U. Picardie Jules Verne (UPJV)
Amiens, France

Florence Levé
MIS, U. Picardie Jules Verne (UPJV)
Amiens, France

{nicolas, mathieu, richard, florence}@algonus.fr

ABSTRACT

Voice and stream segmentation algorithms group notes from polyphonic data into relevant units, providing a better understanding of a musical score. Voice segmentation algorithms usually extract voices from the beginning to the end of the piece, whereas stream segmentation algorithms identify smaller segments. In both cases, the goal can be to obtain mostly monophonic units, but streams with polyphonic data are also relevant. These algorithms usually cluster contiguous notes with close pitches. We propose an independent evaluation of four of these algorithms (Temperley, Chew and Wu, Ishigaki *et al.*, and Rafailidis *et al.*) using several evaluation metrics. We benchmark the algorithms on a corpus containing the 48 fugues of *Well-Tempered Clavier* by J. S. Bach as well as 97 files of popular music containing actual polyphonic information. We discuss how to compare together voice and stream segmentation algorithms, and discuss their strengths and weaknesses.

1. INTRODUCTION

Polyphony, as opposed to monophony, is a music created by simultaneous notes (see Figure 1) coming from several instruments or even from a single polyphonic instrument, such as the piano or the guitar. Polyphony usually implies chords and harmony, and sometimes counterpoint when the melody lines are independent.

Voice and stream segmentation algorithms group notes from polyphonic symbolic data into layers, providing a better understanding of a musical score. These algorithms make inference and matching for relevant patterns easier. They are often based on perceptive rules as studied by Huron [7] or Deutsch [6]. Chew and Wu gathered these rules into four principles [2]:

- (*p1*) Voices are monophonic;

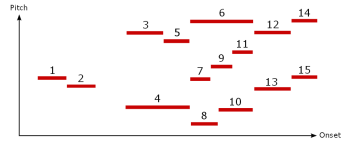


Figure 1: In this piano-roll representation, each segment describes a note. The horizontal axis represents time and the vertical axis represents the pitch.

- (*p2*) At least once, all voices must be played simultaneously;
- (*p3*) Intervals are minimized between successive notes in the same stream or voice (pitch proximity);
- (*p4*) Voices tend not to cross.

Voice segmentation algorithms extract voices from the beginning to the end of the piece. Usually, the resulting voices are monophonic (*p1*) and, at some point, all the voices do appear (*p2*). The algorithms described by Chew and Wu [2] and Ishigaki *et al.* [9] first identify *contigs* of notes, then link these contigs. These algorithms will be discussed later. De Valk *et al.* [5] proposed a machine learning model with a neural network to separate voices in lute tablatures. The study of Kirilin and Utgoff [13] uses another machine learning model to separate voices, taking in consideration both actual polyphony and *implicit* polyphony, such as the one obtained with arpeggios.

Stream segmentation algorithms identify segments generally smaller than complete voices. A *stream* is a group of coherent notes, usually respecting principles such as *p3* and *p4*. Temperley's algorithm [17] extracts streams with respect to several constraints. Rafailidis *et al.*'s algorithm [16], based on an earlier work by [11], uses a *k*-nearest neighbors clustering technique on individual notes. Both algorithms will be discussed in Sections 3.1 and 3.2. The study by Madsen and Widmer [15], inspired by Temperley [17], allows crossing voices. The method of Kilian and Hoos [12] starts by cutting the input score into sections called *slices* such that all the notes of a slice overlap; Then, an optimization method involving several evaluation

© Nicolas Guiomard-Kagan, Mathieu Giraud, Richard Groult, Florence Levé.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Nicolas Guiomard-Kagan, Mathieu Giraud, Richard Groult, Florence Levé. "Comparing Voice and Stream Segmentation Algorithms", 16th International Society for Music Information Retrieval Conference, 2015.

functions is applied to divide and combine the slices into voices; The output voices can contain chords.

Depending on the algorithms, the predicted streams can thus be small or large. However, such algorithms do not predict groups of notes, especially contiguous relevant notes, and may thus be compared against full voice segmentation algorithms. De Nooijer *et al.* [4] made a comparison by humans of several voice and stream separation algorithms for melody finding.

In this paper, we independently evaluate some of these algorithms, benchmarking in the same framework voice and stream segmentation algorithms. We compare some simple and efficient algorithms that were described in the literature [2, 9, 16] and added the algorithm in [17] for which an implementation was freely available. Our corpus includes Bach's fugues (on which many algorithms were evaluated) but also pop music containing polyphonic material made of several monophonic tracks. The next two sections detail these algorithms. Section 4 presents the evaluation corpus, code, and methods. Section 5 details the results and discusses them.

2. VOICE SEPARATION ALGORITHMS

2.1 Baseline

To compare the different algorithms, we use a very simple reference algorithm, based on the knowledge of the total number of voices ($p2$). The *baseline algorithm* assigns a reference pitch for each voice to be predicted, then assigns each note to the voice which has the closest reference pitch (Figure 2).



Figure 2: The baseline algorithm assigns each note to the voice having the closest reference pitch. This reference pitch is computed by averaging pitches on segments having the highest number of simultaneous notes. Here the middle voice, Voice 1, has a reference pitch that is the average of the pitches of notes 7, 9 and 11.

2.2 CW

The CW algorithm separates voices by using the four principles ($p1$, $p2$, $p3$, $p4$) [2].

Contigs. The first step splits the input data into blocks such that the number of notes played at the same time during one block does not change. Moreover, when a note

crosses the border of two blocks and stops or starts to sound inside a block, the block is split in two at this time. The obtained blocks are called *contigs* (Figure 3). By construc-

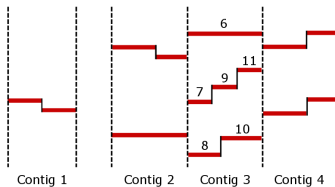


Figure 3: Four contigs: Contig 3 contains three fragments, $\{6\}$, $\{7, 9, 11\}$ and $\{8, 10\}$.

tion, the number of played notes inside a contig is constant. Notes are grouped from the lowest to the highest pitch in *voice fragments* (Figure 3).

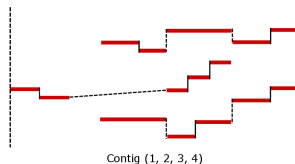


Figure 4: Connection Policy: All fragments are connected with respect to $p3$.

Connection Policy. The second step links together fragments from distinct contigs (see Figure 4). The contigs containing the maximal number of voices are called *maximal voice contigs* ($p2$). The connection starts from these maximal contigs: Since the voices tend not to cross, the order of the voices attributed to fragments of these contigs has a strong probability to be the good one ($p2$ and $p4$).

Given two fragments in contiguous contigs, CW defines a *connection weight*, depending on n_1 , the last note of the left fragment, and on n_2 , the first note of the right fragment. If n_1 and n_2 are two parts of the same note, this weight is $-K$, where K is a large integer, otherwise the weight is the absolute difference between the pitches of the two notes ($p3$). The fragments connected between two contigs are the ones which minimize the total connection weight (Figure 5).

2.3 CW-Prioritized

Ishigaki *et al.* [9] proposed a modification of CW algorithm in the merging step between the contigs. Their key observation is that the *entry of a voice* is often non-ambiguous, contrary to the exit of a voice that can be a "fade out" which is difficult to precisely locate. Instead of starting from maximal voice contigs, they thus choose to start

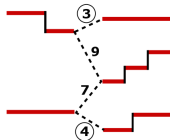


Figure 5: Connection between contigs: The selected links are the ones minimizing the total weight ($3 + 4 = 7$).

only from adjacent contigs with an *increasing* number of voices. For example in Figure 3, the procedure starts by merging Contig 1 with Contig 2. The choice of merged fragments is identical to the method described in CW algorithm. After the merge of all fragments of two adjacent contigs c_1 and c_2 , we get a new contig containing the same number of voices than in c_2 (see Figure 6).

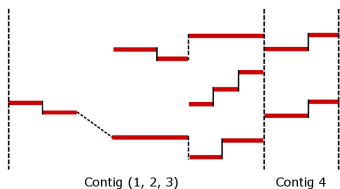


Figure 6: Contig combining: Contigs 1, 2, then 3 are combined, resulting in a Contig 1+2+3 with 3 voices.

The procedure described above is reiterated as long as two adjacent contigs have an increasing number of voices. If at the end of this procedure, there is more than one contig, they are merged by the original CW connection policy.

3. STREAM SEGMENTATION ALGORITHMS

We also study *stream segmentation* algorithms, which do not segment a score into voices but into streams that may include overlapping notes. Streams can be melodic fragments, but also can cluster related notes, such as chords. A voice can be thus split into several streams, and a stream can cluster notes from different voices.

3.1 Streamer

The algorithm proposed by Temperley extracts streams while respecting several constraints [17]. The first constraint is pitch proximity: two contiguous notes with close pitches are placed in the same stream ($p3$). The second constraint is temporal: when there is a long rest between two notes, the second note is put into a new stream (Figure 7). The last principle allows the duplication of a note in two voices (provided that the streams do not cross, $p4$).

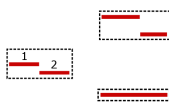


Figure 7: Due to the rest after note 2, Streamer assigns notes 1 and 2 to a stream that does not include any other notes.

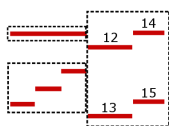


Figure 8: Stream Segment assigns notes 12, 13, 14, and 15 in a same stream. The notes 13-15 can be seen as a transposition of notes 12-14, forming a succession of chords.

3.2 Stream Segment

The algorithm by Rafailidis *et al.* [16] clusters notes based on a k -nearest-neighbors clustering. The algorithm first computes a distance matrix, which indicates for each possible pair of notes whether they are likely to belong to the same stream. The distance between two notes is computed according to their synchronicity (Figure 8), pitch and onset proximity (among others criteria); then for each note, the list of its k -nearest neighbors is established.

3.3 CW-Contigs

We finally note that the first step of the CW algorithm (contig creation) can be considered as a stream segmentation algorithm. We call this first step CW-Contigs. For example, on the Figure 3, this method creates 8 streams corresponding to the 8 voice fragments of the four contigs.

4. EVALUATION CORPUS AND METRICS

4.1 Evaluation corpus

Usually these algorithms are evaluated on classical music, in particular on counterpoint music such as *fugues*, where the superposition of melodic lines gives a beautiful harmony. As a fugue is made up several voices, this naturally constitutes a good benchmark to evaluate voice separation algorithms [2, 5, 9–11, 15–17]. We thus evaluated the algorithms on the 48 fugues of the two books of the *Well-Tempered Clavier* by J.-S. Bach¹.

We also wanted to evaluate other forms of polyphonic writing. The problem is to have a ground truth for this task. From a set of 2290 MIDI files of popular music, we formed a corpus suitable for the evaluation of these algorithms. We focused on MIDI tracks (and not on MIDI channels). We kept only “monophonic” tracks (where at most one note is played at any time) of sufficient length (at least 20 % of the length of the longest track). We deleted the tracks corresponding to the drums. We considered each remaining

¹ .krn files downloaded from kern.ccarh.org [8]

corpus	wic-i	wic-ii	pop
files	24	24	97
voices	3.5	3.4	3.0
notes	1041	1071	874

Table 1: Files, and average number of voices and notes for each corpus.

track as an independent voice. Finally, we kept 97 MIDI files with at least 2 voices, composed on average of 3.0 voices (see Table 1).

4.2 Evaluation code

We implemented the algorithms CW-Contigs, CW-CW-Prioritized and Stream Segment, using a Python framework based on music21 [3]. The Streamer algorithm² was run with default parameters. As it quantizes input files, the offset and duration of notes in the output are slightly different from the ones in our original files: We thus had to associate notes to the correct ones.

4.3 Evaluation metrics

4.3.1 Note-based evaluation.

A first evaluation is to ask whether the voices are correctly predicted. The *note precision (NPR)* is the ratio between the number of notes correctly predicted (in the good voice) over the total number of notes. On one voice, this measure is the same than the *average voice consistency (AVC)* defined by [2]. However on a piece or on a corpus, we compute the ratio on the total number of notes, instead of averaging ratios as in [2]. Especially in the pop corpus, the distribution of notes is not equal in all pieces and all voices, and this measure better reflects the ability of the algorithm to assign the good voice to each note.

Computing NPR requires to assert *which voice in the prediction corresponds to a given voice of the ground truth*. In a fugue, there may be a formal way to exactly define the voices and number them, from the lowest one to the highest one. But, in the general case, this correspondance is not always obvious. By construction, the two voice segmentation algorithms studied here predict a number of voices equal to the maximal number of voices, whereas the stream segmentation algorithms have no limit for the number of streams. In the general case, one solution is to compare each voice predicted by the algorithm with the *most similar voice of the ground truth*, for example taking the voice of the ground truth sharing the highest number of notes with the predicted voice.

Note-based evaluation tends to deeply penalize some errors in the middle of the scores: When a voice is split in two, half of the notes will be counted as false even if the algorithm did “only one” mistake. Moreover, this is not

a fair way to evaluate stream segmentation algorithms, as they may predict (many) more streams than the number of voices. We thus use two other metrics, that better measure the ability of the algorithms to gather notes into voices, even when a voice of the ground truth is mapped to several predicted voices. These metrics do not require to make the correspondence between predicted voices and voices of the truth.

4.3.2 Transition-based evaluation.

The result of voice or stream segmentation methods can be seen as a set of *transitions*, that are pairs of successive notes in a same predicted voice or stream. We compare these transitions against the transitions defined by the ground truth, and compute usual precision and recall ratios.

The *transition precision (TR-prec)* (called *soundness* by [13]) is the ratio of correctly assigned transitions over the number of transitions in the predicted voices. This is related to *fragment consistency* defined in [2] – but the fragment consistency takes only into account the links between the contigs, and not all the transitions. The *transition recall (TR-rec)* (called *completeness* by [13]) is the ratio of correctly assigned transitions over the number of transitions in the truth. This is again related to *voice consistency* of [2].

For each piece, we compute these ratio on all the voices – taking the number of correct transitions inside *all* the voices, and computing the ratio over the number of transitions inside *either all* the predicted voices or *all* the truth. When the number of voices in the ground truth and in the prediction are equal, the TR-prec and TR-rec ratios are thus equal: we simply call this measure *TR*. Figure 12, at the end of the paper, details an example of NPR and TR values for the six algorithms.

4.3.3 Information-based evaluation.

Finally, we propose to adapt techniques proposed to evaluate music segmentation, seen as an assignation of a label to every audio (or symbolic) frame [1, 14]. Lukashevich defines two scores, S_o and S_u , based on normalized entropy, reporting how an algorithm may over-segment (S_o) or under-segment (S_u) a piece compared to the ground truth. The scores reflect how much information there is in the output of the algorithm compared to the ground truth (S_o) or conversely (S_u). Here, we use the same metrics for voice or stream segmentation: both the ground truth and the output of any algorithm can be considered as an assignation of label to every note. On the probability distribution of these labels, we then compute the entropies $H(\text{predicted}|\text{truth})$ and $H(\text{truth}|\text{predicted})$, that become S_o and S_u after normalization [14]. As these scores are based on notes rather than on transitions, they enable to measure whether the clusters are coherent, even in situations when two simultaneous voices are merged in a same stream (giving thus bad TR ratios).

² downloaded from www.link.cs.cmu.edu/melisma

	wtc-i					wtc-ii					pop				
	avg.	NPR	TR	S_o	S_u	avg.	NPR	TR	S_o	S_u	avg.	NPR	TR	S_o	S_u
Baseline	3.5	71.4%	63.7%	0.48	0.48	3.4	71.9%	62.6%	0.45	0.45	3	89.5%	87.1%	0.77	0.75
CW	3.5	83%	95.9%	0.73	0.73	3.4	87.8%	95.6%	0.73	0.73	3	84.6%	88.7%	0.76	0.76
CW-Prioritized	3.5	82.5%	97.4%	0.72	0.72	3.4	86.5%	97.1%	0.74	0.74	3	64.8%	89.4%	0.51	0.5
	avg.	TR-prec	TR-rec	S_o	S_u	avg.	TR-prec	TR-rec	S_o	S_u					
Streamer	16	75.6%	68.3%	0.46	0.62	15.4	75.6%	65.2%	0.42	0.57					
Stream Segment	191.1	76.5%	62.1%	0.23	0.79	214	77.4%	61.9%	0.21	0.79					
CW-Contigs	226.2	99.4%	86.7%	0.34	0.98	282.3	99.4%	86.8%	0.34	0.98					

Table 2: Results on the fugues and on the pop corpora. “avg.” is the average number of voices or streams predicted.

5. RESULTS AND DISCUSSION

We evaluated the six algorithms on the 48 fugues of *Well-Tempered Clavier* by J. S. Bach, and moreover the voice separation algorithms were evaluated on the 97 pop files. Table 2 details the results.

5.1 Results

Note and transition-based evaluation. Between 80 % and 90 % of the notes are assigned correctly to the right voice by at least one of the voice separation algorithms. The results confirm that these NPR metric is not very meaningful. The baseline has good NPRs, and on the pop corpus, the baseline NPR is even better than CW and CW-Prioritized. Compared to the baseline algorithm, all algorithms output longer fragments (see Figure 9). As expected, the transition ratio (TR) metrics are better to benchmark the ability of the algorithms to gather relevant notes in the same voice: all the algorithms have better TR metrics than the baseline.

The three stream segmentation algorithms predict more streams than the number of voices in the ground truth, hence low TR-rec ratios. The TR-prec ratios are higher, better than the baseline, and the CW-Contigs has an excellent TR-prec ratio.

Information-based evaluation. An extreme case is perfect prediction, with $\text{NPR} = \text{TR} = 100\%$, $S_o = 1$ and $S_u = 1$ (like in Bach’s Fugue in E minor BWV 855 for both CW and CW-Prioritized). In a pop song (allsaints-bootiecall) where two voices play mostly same notes, the baseline algorithm merges all notes in the same voice, so NPR and TR are close to 50%, but S_o is close to 1 and S_u close to 0.



Figure 9: Notes attributed to the wrong voice with the baseline (left) and CW (right) algorithms on Bach’s Fugue #2 – book II (in C minor, BWV 871). When CW makes errors, the voices are kept in a same predicted voice.

In the general case, S_u is correlated with TR-prec, and S_o with TR-rec. As expected, in stream segments algorithms, S_u is better than S_o . Note that the Stream Segment has not the best TR-prec ratio (sometimes, it merges notes that are in separate voices), but it has a quite good S_u score among all the algorithms (when it merges notes from separate voices, it tends to put in the same stream all notes that are in related voices). The best S_u scores are obtained by the CW-Contigs, confirming the fact that the contig creation is a very good method that makes almost no error.

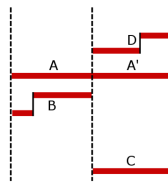


Figure 10: A note spanning two contigs is split in A and A' . CW and CW-Prioritized link the fragments $(A + A')$, keeping A in the same voice. The original implementation of Ishigaki *et al.* links the fragments $(A + D)$, (B, A') , duplicating the whole note $A + A'$.

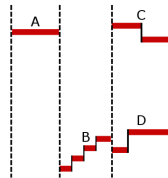


Figure 11: Fragments A and B are in different contigs due to the overlap of previous notes. Both CW-Prioritized and the original implementation of Ishigaki *et al.* link the fragments $(A + B + D)$ and (C) , whereas CW links the fragments $(A + C)$ and $(B + D)$.

5.2 Partitioned notes and link weights

With CW algorithm, when a note is cut between two contigs and the voices assigned to those two fragments are different, the predicted voices contain more notes than in the input data. This case was not detailed in the study [2]. To avoid split notes in the output of the algorithm, we choose to allow voice crossing exactly at these points (Figure 10).

Our results for CW-Prioritized differ from the ones obtained in [9]: Their AVC was better compared to CW. In our implementation, the NPR ratio is lower for CW-Prioritized compared to CW. In our implementation (as in the

original algorithm of CW), there is a $-K$ weight to the link between two parts of the same note. In the Ishigaki *et al.* implementation, this weight is -1 , and thus the algorithm keeps partitioned notes in the output (see Figure 10). Despite this difference, our CW-Prioritized implementation gives good results by considering TR both on the fugues and on the pop corpus, even if it merges incorrectly some contigs (see Figure 11).

5.3 A challenging exposition passage in a fugue

Figure 12 shows the results of the algorithms on an extract of Bach's Fugue #16 – book I. This passage is quite challenging for voice separation: all the four voices enter in succession, and there is a sixth interval in the head of the subject that often put voices very close. In the last measure of the fragment, there is even a crossing of voices when the soprano is playing this sixth interval.

The algorithms behave differently on this passage, but none of them perfectly analyze it. Only CW-Prioritized predicts correctly the first three measures, especially the start of the alto voice at the first two beats of measure 12. CW selects a bad link at the third beat of measure 14, resulting in a bad prediction in measures 12/13/14 (but a high TR ratio overall). Except on the places where almost all the algorithms fail, Streamer has a correct result. Streamer creates many more streams, and, as expected, assigns notes that overlap in the same stream, as on the first beat of measure 12.

Finally, none of the algorithms successfully handles the voice crossing, measure 15. CW-Contigs made here its only clustering error (otherwise it has an excellent TR-prec), linking the D of the soprano with the following G of the alto. As expected, this error is found again in CW and CW-Prioritized, and Streamer also splits apart the notes with the highest pitch from the notes with the lowest pitch. At this point, Stream Segment creates streams containing both voices. Handling correctly this passage would require to have a knowledge of the patterns (including here the head of the subject with the sixth leap) and to favor to keep these patterns in a same voice, allowing voice crossing.

6. CONCLUSIONS

Both voice and stream segmentation methods cluster notes from polyphonic scores into relevant units. One difficulty when benchmarking such algorithms is to define a ground truth. Beside the usual fugues corpus, we proposed some ideas to establish a pop corpus with polyphonic data suitable for evaluation.

Even stream segmentation algorithms give good results in separating voices, as seen by the TR ratios and the S_u score. The Streamer algorithm is very close to a full voice separation, predicting monophonic streams. The Stream

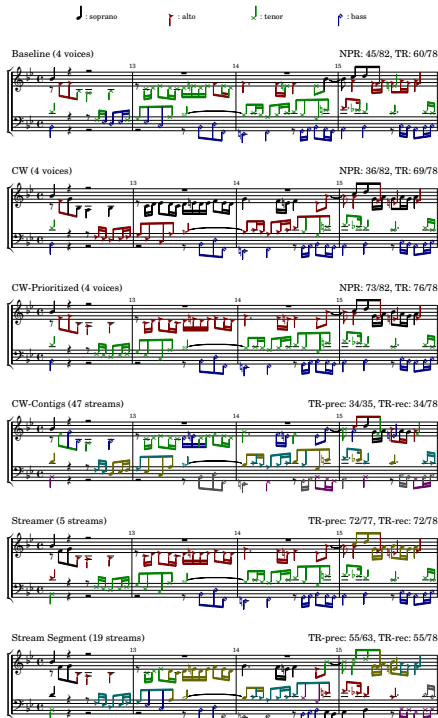


Figure 12: Output of the five algorithms on the measures 12 to 15 of Bach's Fugue #16 – book I (in G minor, BWV 861). After the initial chord with almost all the voices, the voices enter in succession (alto and tenor: m12, bass: m13, soprano: m15).

Segment algorithm further enables to output some polyphonic streams that may be relevant for the analysis of the score.

Focusing on voice separation problem, the contig approach, as initially proposed by [2], seems to be an excellent approach – very few transition errors are made inside contigs, as shown by the raw results of the CW-Contigs algorithm. The challenge is thus to do the right connections between the contigs. The ideas proposed by [9] are interesting. In our experiments, we saw a small improvement in our CW-Prioritized implementation compared to CW, but details on how partitioned notes are processed should be handled carefully. Further research should be done to improve again the contig connection.

7. REFERENCES

- [1] Samer Abdallah, Katy Noland, Mark Sandler, Michael A Casey, Christophe Rhodes, et al. Theory and evaluation of a bayesian music structure extractor. In *International Conference on Music Information Retrieval (ISMIR 2005)*, pages 420–425, 2005.
- [2] Elaine Chew and Xiaodan Wu. Separating voices in polyphonic music: A contig mapping approach. In *International Symposium on Computer Music Modeling and Retrieval (CMMR 2005)*, pages 1–20. Springer, 2005.
- [3] Michael Scott Cuthbert and Christopher Ariza. music21: A toolkit for computer-aided musicology and symbolic music data. In *International Music Information Retrieval Conference (ISMIR 2010)*, 2010.
- [4] Justin de Nooijer, Frans Wiering, Anja Volk, and Hermi JM Tabachneck-Schijf. An experimental comparison of human and automatic music segmentation. In *International Computer Music Conference (ICMC 2008)*, pages 399–407, 2008.
- [5] Reinier de Valk, Tillman Weyde, and Emmanouil Benetos. A machine learning approach to voice separation in lute tablature. In *International Society for Music Information Retrieval Conference (ISMIR 2013)*, pages 555–560, 2013.
- [6] Diana Deutsch. Grouping mechanisms in music. *The psychology of music*, 2:299–348, 1999.
- [7] David Huron. Tone and voice: A derivation of the rules of voice-leading from perceptual principles. *Music Perception*, 19(1):1–64, 2001.
- [8] David Huron. Music information processing using the Humdrum toolkit: Concepts, examples, and lessons. *Computer Music Journal*, 26(2):11–26, 2002.
- [9] Asako Ishigaki, Masaki Matsubara, and Hiroaki Saito. Prioritized contig combining to segregate voices in polyphonic music. In *Sound and Music Computing Conference (SMC 2011)*, volume 119, 2011.
- [10] Anna Jordanous. Voice separation in polyphonic music: A data-driven approach. In *International Computer Music Conference (ICMC 2008)*, 2008.
- [11] Ioannis Karydis, Alexandros Nanopoulos, Apostolos Papadopoulos, Emiliios Cambouropoulos, and Yannis Manolopoulos. Horizontal and vertical integration/segregation in auditory streaming: a voice separation algorithm for symbolic musical data. In *Sound and Music Computing Conference (SMC 2007)*, 2007.
- [12] Jürgen Kilian and Holger H Hoos. Voice separation—a local optimization approach. In *International Conference on Music Information Retrieval (ISMIR 2002)*, 2002.
- [13] Phillip B Kirlin and Paul E Utgoff. VOISE: learning to segregate voices in explicit and implicit polyphony. In *International Conference on Music Information Retrieval (ISMIR 2005)*, pages 552–557, 2005.
- [14] Hanna M Lukashevich. Towards quantitative measures of evaluating song segmentation. In *International Conference on Music Information Retrieval (ISMIR 2008)*, pages 375–380, 2008.
- [15] Søren Tjagvad Madsen and Gerhard Widmer. Separating voices in midi. In *International Conference on Music Information Retrieval (ISMIR 2006)*, pages 57–60, 2006.
- [16] Dimitris Rafailidis, Alexandros Nanopoulos, Emiliios Cambouropoulos, and Yannis Manolopoulos. Detection of stream segments in symbolic musical data. In *International Conference on Music Information Retrieval (ISMIR 2008)*, 2008.
- [17] David Temperley. *The Cognition of Basic Musical Structures*. Number 0-262-20134-8. Cambridge, MA: MIT Press, 2001.

MELODY EXTRACTION BY CONTOUR CLASSIFICATION

Rachel M. Bittner¹, Justin Salamon^{1,2}, Slim Essid³, Juan P. Bello¹

¹ Music and Audio Research Lab, New York University

² Center for Urban Science and Progress, New York University

³ Télécom Paris-Tech

rachel.bittner@nyu.edu

ABSTRACT

Due to the scarcity of labeled data, most melody extraction algorithms do not rely on fully data-driven processing blocks but rather on careful engineering. For example, the Melodia melody extraction algorithm employs a pitch contour selection stage that relies on a number of heuristics for selecting the melodic output. In this paper we explore the use of a discriminative model to perform purely data-driven melodic contour selection. Specifically, a discriminative binary classifier is trained to distinguish melodic from non-melodic contours. This classifier is then used to predict likelihoods for a track’s extracted contours, and these scores are decoded to generate a single melody output. The results are compared with the Melodia algorithm and with a generative model used in a previous study. We show that the discriminative model outperforms the generative model in terms of contour classification accuracy, and the melody output from our proposed system performs comparatively to Melodia. The results are complemented with error analysis and avenues for future improvements.

1. INTRODUCTION

Melody extraction has a variety of applications in music retrieval, classification, transcription and analysis [15]. A precise definition of melody that takes into account all possible scenarios has proven elusive for the MIR community. In this paper we consider two different definitions of melody [1]: The f_0 curve of the predominant melodic line drawn from a single source (melody type 1), and the f_0 curve of the predominant melodic line drawn from multiple sources (melody type 2).


Some approaches to melody extraction are source separation-based [4, 18], first isolating the melodic source from the background and then tracking the pitch of the resulting signal. The most common approaches are based on the notion of salience [3, 7, 13, 14], and are variants of the following steps (1) audio pre-processing, (2) salience

function computation, (3) f_0 tracking, and (4) voicing decisions. Steps (3) and (4) for these methods are each based on a series of carefully chosen heuristic steps, and are limited to the data they were designed for. A recent trend in Music Information Retrieval research is to combine domain knowledge with data driven methods [8], using domain informed feature representations as input to data-driven models. To the best of our knowledge, only one melody extraction approach [5] has been proposed to date using a fully data driven method. However, the features employed were poor for the task (magnitude Fourier coefficients), and used only limited temporal modeling via HMM smoothing. Additionally, at the time, only a small amount of data was available. The recent availability of annotated melody data allows for new exploration into data driven methods for melody extraction.

In this paper, we present a system for melody extraction which replaces the common series of heuristic steps with a data-driven approach. We propose a method for scoring extracted contours (short, continuous pitch sequences) using a discriminative classifier, and a Viterbi-based method for decoding the output melody. We show that our method performs competitively with Melodia [14]. The implementation of the proposed method and the code used for each experiment is available on Github¹. The remainder of this paper is organized as follows: in Section 2 we give an overview of Melodia; Section 3 describes our proposed method for melody extraction; in Section 4 we present experiments evaluating the effectiveness of our method, including a comparison with Melodia, and in Section 5 we discuss the conclusions and avenues for future work.

2. MELODIA

Melodia [14], a salience-based melody extraction algorithm, has proved to perform particularly well. The algorithm is comprised of four processing blocks: sinusoid extraction, salience function computation, contour creation and characterization, and finally melody selection. In the first block, spectral peaks are detected, and precise frequencies of each component are estimated using their instantaneous frequency. In the second stage a harmonic summation-based salience function is computed. In the third block, the peaks of the salience function are tracked

 © Rachel M. Bittner, Justin Salamon, Slim Essid, Juan P. Bello. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Rachel M. Bittner, Justin Salamon, Slim Essid, Juan P. Bello. “Melody Extraction by Contour Classification”, 16th International Society for Music Information Retrieval Conference, 2015.

¹ www.github.com/rabitt/contour_classification

into continuous pitch contours using auditory streaming cues. Additionally, a number of features are computed for each contour:

- Duration (seconds)
- Pitch mean and standard deviation (in cents)
- Saliency mean, standard deviation, and sum
- Vibrato presence (binary), rate (Hz), extent (cents), coverage (fraction of contour with vibrato)

The melody f_0 trajectory is obtained in the fourth block by filtering out non-melodic contours based on their features in combination with an iterative estimation of the global melodic trend. This step exploits the contour feature distributions to perform the filtering, but does so in a heuristic fashion. For further details the reader is referred to [14].

Recently, Melodia was evaluated on the MedleyDB [1] dataset which contains considerably more variety in musical style than previous datasets. The results were shown to be substantially lower than for the existing datasets. In particular, it was reported that Melodia’s performance on music with vocal melodies was better than on music with instrumental melodies. This indicates that the heuristic steps at the contour selection stage may be well tuned for singing voice, but less so for instrumental melodies. Since these heuristics are hard coded, the algorithm cannot be adjusted for different kinds of data or different concepts of melody. We will show that these steps can be replaced by a data driven approach.

In [16], Salamon, Peeters, and R obel proposed to replace the heuristic melodic selection block of Melodia with a generative classification model. The contour features were used to estimate two multivariate Gaussian distributions, one for melodic and another for non-melodic contours. These distributions were used to define the “melodiness” score, computed as the likelihood ratio of the two distributions.

The final f_0 sequence was obtained by taking the f_0 of the contour with the highest “melodiness” at each frame. The authors showed that the generative model could produce similar (albeit not equally good) results in terms of pitch accuracy, but the model lacked a voicing detection step. This was addressed by combining the model with the voicing detection filter of the original Melodia algorithm.

Finally, in [17] the authors combined Melodia’s contour features with additional features to train a discriminative model for classifying different musical genres. Their experiments showed that the contour features carry discriminative melodic information. This outcome, together with that of [16] and the release of MedleyDB, gives compelling motivation for the exploration of discriminative models using pitch contour features for solving the problem of melodic contours selection.

3. METHOD

The proposed system uses the pitch contours and contour features generated by Melodia ². The method consists of

² These are taken from intermediate steps in the Vamp plugin’s implementation

a contour labeling stage, a training stage where a classifier is fit to discriminate melody from non-melody contours, and a decoding stage which generates a final f_0 sequence. Melody output is computed using a trained classifier as shown in Figure 1.

3.1 Contour Labeling

To generate contours for a musical audio signal, we use the first three processing blocks of the Melodia algorithm directly (see [14] for details). Each contour is represented by a sequence of tuples (time, frequency, saliency). As described in Section 2, the third block also computes a set of descriptive features for each contour, which we use to train the model in Section 3.2.

During training, extracted contours are assigned binary labels: 1 if the contour should be considered as a part of the melody and 0 otherwise. The labels are chosen by comparing the amount of overlap between each contour and the ground truth annotation. Given an annotation $a(t)$ with $0 \leq t \leq T$, a contour $c(t)$ spanning the time interval $t_1 \leq t \leq t_2$ is compared with $a(t)$ over the time range $t_1 \leq t \leq t_2$. The amount of overlap between these two sequences is computed using “Overall Accuracy” [15], defined as:

$$\text{Acc}_{ov} = \frac{1}{L} \sum_{i=0}^{L-1} v_i \mathcal{T} [| \hat{\varphi}_i - \varphi_i | + (1 - v_i)(1 - \hat{v}_i)] \quad (1)$$

where L is the number of reference/estimate examples, v_i and \hat{v}_i are the (binary) voicings of the reference and estimate respectively, φ_i and $\hat{\varphi}_i$ are the f_0 values in cents of the reference and estimate respectively, and \mathcal{T} is a threshold function equal to 1 if the argument is less than 0.5, and 0 otherwise. Given a minimum overlap threshold α , if $\text{Acc}_{ov} > \alpha$ the contour is labeled as melody. Note that if $\alpha = 1$, because of the strict inequality, all contours would be labeled as non-melody. Despite containing extraneous information, a contour with a small degree of overlap still contains part of the melody. Labeling it as non-melody removes any possibility of the melody-portion of the contour ending up in the final extracted melody (i.e., lower recall). On the other hand, labeling it as melody potentially results in having non-melody information included in the melody (i.e., lower precision). Thus, there is an inherent trade-off between melody precision and recall based on the value of the overlap threshold α .

3.2 Contour Classification

We normalize the features per track to remove variance caused by track-level differences. The saliency features are each divided by the maximum saliency value in the track to remove differences based on overall track saliency. The duration feature is normalized so that across the track the minimum value is 0 and the maximum value is 1. The feature “total saliency” is additionally re-scaled to reflect the normalized duration.

These features and the computed labels are used to train a random forest classifier [2]. We use the random forest implementation in `scikit-learn` [11] with 100 trees and

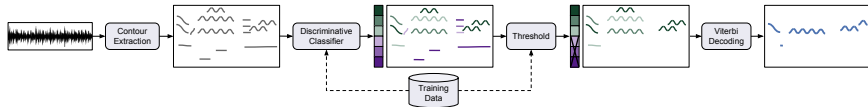


Figure 1. Block diagram of the proposed system (left to right): pitch contours are extracted from an audio signal, a classifier is used to score the contours and remove those below a threshold, the final f_0 sequence is obtained using Viterbi decoding.

choose the maximum depth parameter by cross validating over the training set. In our experiments, the classifier was trained with roughly 11,000 examples for melody 1 and roughly 15,000 for melody 2. Because our class distributions tend to be biased towards non-melody examples, the classifier is trained with class weights inverse to the class distributions. Once the classifier is trained, we use it to predict the probability that a given contour is melody. In the case of a random forest, the melody likelihood is computed as the fraction of trees that classify a given example as melody.

3.3 Melody Decoding

We create an output melody by first removing contours whose likelihood falls below a threshold β and then decoding to generate a final melody. The thresholding step is necessary because there may be regions of time where only non-melody contours are present. Since decoding only chooses the best path through available contours, having regions with contours which are all non-melody would result in false positives. Aside from the contour extraction, the choice of this threshold is the single most important step for determining the voicing of the output melody.

This raises the question: what is the best way to determine the likelihood threshold β ? A natural choice is $\beta = 0.5$, as this is the threshold that has been optimized by the machine for maximum classification accuracy. While this threshold gives us nearly perfect precision for the melody class, the recall is extremely low. We instead choose the threshold that yields the highest class-weighted F1 score on a validation set. The chosen value of β in this manner is consistently much lower than 0.5 (typically $\beta \approx 0.1$), resulting in higher recall at the cost of lower precision. It is interesting to note that for our end goal – selecting a single melody sequence – we do not necessarily need perfect precision because false positives can be removed during decoding.

After this filtering step, contours that do not overlap with any other contour are immediately assigned to the melody. The remaining contours have partial overlap with at least one other contour, requiring the melody line to be chosen from within the overlapping segments. Thus, we divide these remaining contours into groups: contours $\{C_1[t], \dots, C_n[t]\}$ each spanning some time interval are assigned to the same group if the union of their intervals forms a contiguous interval.

The path over time through each group of contours is computed using Viterbi decoding [6]. Given a group of n contours, our state space is the set of contour numbers

$\{1, 2, \dots, n\}$. We create a matrix Y of emission probabilities using each contour’s likelihood score $[p_1, p_2, \dots, p_n]$:

$$Y_{it} = \begin{cases} p_i & \text{if } C_i \text{ is active at time } t \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

The transition matrix A , defined to encourage continuity in pitch space, is computed for each group as:

$$A_{ij} = \frac{\sum_{k \neq j} |\log_2(f_i) - \log_2(f_k)|}{(n-1) \sum_{k=1}^n |\log_2(f_i) - \log_2(f_k)|} \quad (3)$$

where f_i is the average frequency (in Hz) of contour i . This transition matrix, simply put, assigns a high transition probability between contours whose (log) frequencies are near one another, and a lower transition probability between contours which are far from one another in log frequency. The prior distribution is set to be uniform. Given the sequence of contour states $S[t]$ computed by Viterbi, for each time point t , the frequency $C_{S[t]}[t]$ is assigned to the melody.

4. EXPERIMENTS

For each of the following experiments we use the MedleyDB Dataset [1]. Of the 122 tracks in the dataset, we use the 108 that include melody annotations. We create train/test splits using an artist-conditional random partition (i.e., tracks from the same artist cannot be in both the train and test set). The complete training set is further split randomly into a training and validation. A given train, validate, and test split contains roughly 78%, 7%, and 15% respectively of the 108 tracks. We repeat each experiment with five different randomized splits to get a sense of the variance of the results when using different data. In Figures 2, 3, and 4, vertical lines indicate the standard deviation over the five splits. Recall that we consider two definitions of melody (Section 1). Consequently, when we report scores for melody type 1, the classifier was trained using the melody 1 annotations, and likewise for melody type 2. All evaluation metrics were computed using `mir_eval` [12].

4.1 Experiment 1: Generative vs. Discriminative Contour Classification

Before evaluating components of the proposed system, we first examine the recall of Melodia’s contour extraction on this dataset. That is, given all extracted contours, what is the percentage of the reference melody that is covered by the contours (in terms of pitch overlap)? We tested this by selecting the “oracle” (i.e., best possible) f_0 curve from the

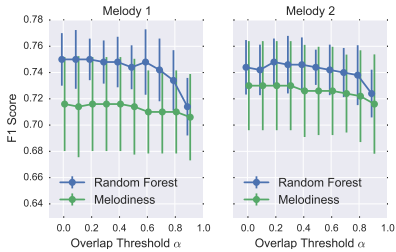


Figure 2. Maximum F1 score achieved per overlap threshold α by the generative and discriminative models.

contours. The oracle output yielded an average Raw Pitch Accuracy of 0.66 ($\sigma = 0.22$) for melody type 1, and 0.64 ($\sigma = 0.20$) for melody type 2. Thus, the best raw pitch accuracy we (or Melodia) could hope to achieve on this dataset is upper bounded by these numbers.

Acknowledging the existence of this glass ceiling, we compare the generative model for scoring contours with the proposed discriminative model. The features used for the discriminative model are those described in Section 2, while the generative model used only the continuous features (i.e., none of the vibrato features)³. The features for the multivariate Gaussian were transformed using the box-cox transformation as in [16], where the transformation’s parameter λ was estimated from the training set and applied to the testing set.

To evaluate the effectiveness of these two methods, we compare the F1 scores achieved by selecting the optimal likelihood threshold β . Figure 2 shows the best achieved F1 scores on the validation set for the two models. We see that the random forest classifier obtains a better F1 score for all values of α . Interestingly, the F1 score achieved by the multivariate Gaussian is less affected by α than the Random Forest, which decreases as α increases. Note that neither classifier achieves an F1 score above 75%. This suggests that either the models are not complex enough or that the classes are not completely discriminable using this set of features. Since our feature set is relatively small, the latter is likely, and the performance of both of these models would likely benefit from a larger feature set. However, fitting a high dimensional multivariate Gaussian requires a large amount of data. Thus, another advantage of using a random forest classifier is that increasing the dimensionality of the feature space does not necessarily require more data.

One might argue that the difference in performance of the two methods could be due to the fact that the vibrato features are not used in the multivariate Gaussian model. However, a post-hoc analysis of the importance of the vibrato features within the random forest classifier (for melody 1 with $\alpha = 0.5$) showed that they were by a

³ We initially included the vibrato features for the generative model, but the results were extremely poor.

Melody Type	OA	RPA	RCA
1	1.6	2.5	2.5
2	2.4	4.2	2.1

Table 1. Percentage point difference between Viterbi decoding and taking a simple maximum.

large margin the least important features in the set. In fact, the presence of vibrato contributed to discriminating only $\approx 0.03\%$ of the training samples. The most discriminative features for the random forest were the salience standard deviation, followed by pitch mean, followed by pitch standard deviation.

Overall, we see that the random forest consistently outperforms the multivariate Gaussian, and has the additional benefit of scalability to a much larger feature set.

4.2 Experiment 2: Decoding Method

Our second experiment examines the effect of our Viterbi decoding strategy. First, we compare it with an approach based on the one used in [16], where the f_0 value at each point in time was chosen by taking a simple maximum over the “melodiness” score. For our comparison, we take the maximum over the likelihoods produced by the classifier after thresholding.

We found that Viterbi decoding consistently showed an improvement in the melody evaluation metrics on each track. For some particular tracks, Viterbi decoding improved the output by up to 10 percentage points. Table 1 shows the average percentage point increase per track by using Viterbi over the simple maximum. The metrics shown are the Overall Accuracy (OA), Raw Pitch Accuracy (RPA), and Raw Chroma Accuracy (RCA) [15]. We see a particularly good improvement for melody 2, where Viterbi decoding increases the average raw pitch accuracy by more than 4 percentage points.

Figure 3 shows each melody evaluation metrics across the different overlap thresholds α . The values plotted are averages over each of the 5 experiments, where the error bars indicate the standard deviation. Surprisingly, we see very little difference in any of the metrics for both melody types. We saw in Figure 2 that the F1 score decreased as α increased, which implies that unlike what we might expect, the final melody output is not strongly affected by the F1 score. Note, however, that the F1 score is computed on a different set of labels for each value of α . The resilience may be due to the fact that the labels that change as we sweep α are the “noisier” labels, and thus the hardest to classify, whereas the contours that are not affected by the value of α (i.e., very high overlap or no overlap with the annotation) are easier to classify. We conjecture that for each value of α the classifiers are probably performing equally poorly on the noisy contour examples and equally well on the clean examples.

All in all, the deviations in metrics are minor across values of α , and we conclude that the value of α does not have a strong impact on the final melody accuracy. The values

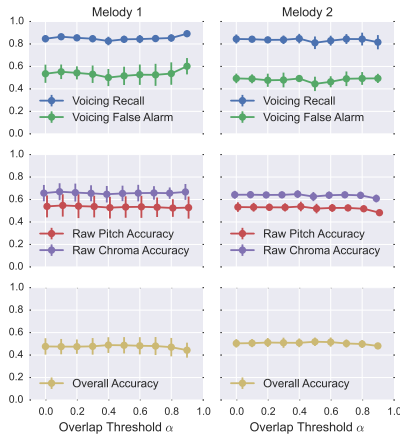


Figure 3. Melody metrics for each overlap threshold α and melody type.

of α that yield the highest scores on the validation set (by a small margin) were $\alpha = 0.5$ for melody 1 and $\alpha = 0.4$ for melody 2, and these values are used for our final system.

4.3 Experiment 3: Melodia vs. Proposed Method

As a final experiment, we compare the proposed method with Melodia. This experiment essentially evaluates the two different contour selection methods, since both methods begin with the same set of contours. Melodia’s parameter ν controls the voicing decision, and is the parameter with the largest impact on the final output. The scores reported for Melodia in this experiment use the value of ν that achieved the best overall accuracy on the training set. The final scores are reported for the test set.

The results for each algorithm are shown in Figure 4. The proposed method performs quite competitively with Melodia. In particular, Melodia only outperforms our system in overall accuracy by 4 percentage points for melody 1 and 2 percentage points for melody 2. The primary metric where the algorithms differ is in the voicing recall and voicing false alarm rates. Our system has significantly better recall than Melodia (33 percentage points higher for melody 1, 9 for melody 2), but also a much higher false alarm rate (34 percentage points higher for melody 1, 14 for melody 2) - in other words, our system assigns contours to the melody much more often than Melodia does.

An interesting example to this point is shown in Figure 5. Both methods achieve the same overall accuracy of ≈ 0.50 , but their output is quite different. Our output gets almost all of the voiced frames correct, but has spurious mistakes outside of the annotation as well. In contrast, Melodia has nearly perfect precision, but misses large segments. This example is characteristic of the difference between the two algorithms – our approach over-predicts

melody, and Melodia under-predicts it. Notice that the proposed method produces spurious frequency values, caused by slight differences in contour start and end points within contour groups. These values could be removed in a future post processing stage.

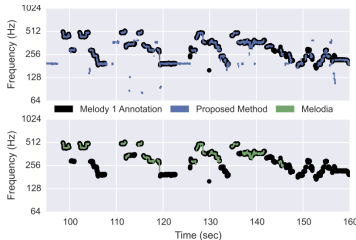


Figure 5. Segment of melody output for “Clara Berry and Wooddog: The Bad Guys”.

This difference is especially significant for tracks containing instrumental melodies. Figure 6 shows a segment from a track with a flute melody. Our approach works quite well for this example, while Melodia misses most of the melodic line. In Figure 4 we also report the overall accuracy for the portions of the data containing vocal (OA-V) and instrumental melodies (OA-I). We see that for instrumental melodies, our method matches Melodia’s performance for melody 1 and slightly outperforms Melodia for melody 2. Conversely, we see the opposite trend for vocals, with Melodia outperforming our method for both melody types. This trend can be largely attributed again to the differences in voicing statistics – vocal melodies in this dataset tend to have many more unvoiced frames than instrumental melodies, so our method’s high false alarm rate hurts our performance for vocal tracks.

Despite the slight difference in metrics, the two algorithms perform similarly, with inversely related pitfalls. It is interesting to note that when the current approach completely fails, so does Melodia. This first and foremost occurs when output from the contour extraction stage is poor, which dooms both methods to failure.

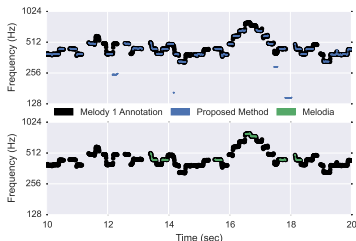


Figure 6. Segment of melody output for “Phoenix: Lark on the Strand/Drummond’s Castle”.

Overall, we see that the proposed method is quite good at correctly choosing melody examples, but the high false

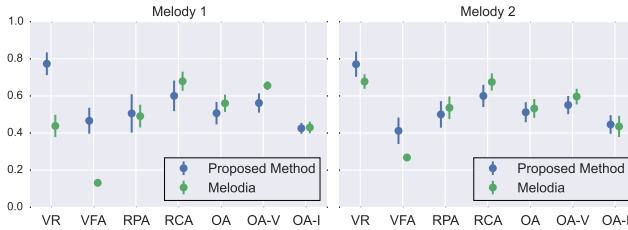


Figure 4. Final melody output scores for the proposed method and Melodia. The metrics are abbreviated on the x-axis as: VR = Voicing Recall, VFA = Voicing False Alarm, RPA = Raw Pitch Accuracy, RCA = Raw Chroma Accuracy, OA = Overall Accuracy, OA-V = Overall Accuracy – vocal tracks, and OA-I = Overall Accuracy – instrumental tracks.

alarm rates hurt its overall scores. This speaks to the classifier’s need for better discrimination between melody and non melody examples. To do this, we need more/better features, a more powerful classifier, or both. This ties back to Ellis and Poliner’s observation in [5]: a large percentage of contours are very easy to distinguish, and the remaining contours are difficult for data driven and heuristic methods alike. This is likely due to the lack of longer time scale features describing the relationship between observations. We as humans are able to distinguish melody from non-melody in a song, but in ambiguous cases, we make our distinction based on what we heard earlier in the song [10].

As a final illustration, Figure 7 shows the output of both algorithms for melody 1 (top) and melody 2 (bottom) for a segment containing a flute and a trumpet. The melody is carried by the flute for most of the track, but in this segment is carried by the trumpet. For melody 1, both methods track the flute, matching the annotation, whereas for melody 2 both methods still track the flute whereas the trumpet line is annotated. Without long-term context giving the algorithm information about which lines have happened previously as background or melody, there is no way for either of these methods to choose the “correct” line.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have shown that replacing Melodia’s heuristic decisions with a series of fully data driven decisions can nearly match Melodia’s overall performance, and there many open avenues for improving our results. In particular, we have shown that a discriminative model can outperform a generative model for labeling contours, and we have provided a detailed evaluation of how each step in the proposed approach influences the final results. Compared to Melodia, we noted that the proposed method has better melody recall, but a considerably worse voicing false alarm rate. To improve the discrimination ability of the classifier, future iterations of this method will first incorporate a wider set of features, including features that describe neighboring contours (octave duplicates, etc.), and features that describe a contour’s relationship with the rest of the track on a longer time scale, potentially including timbre similarity. Additionally, since we are using a rel-

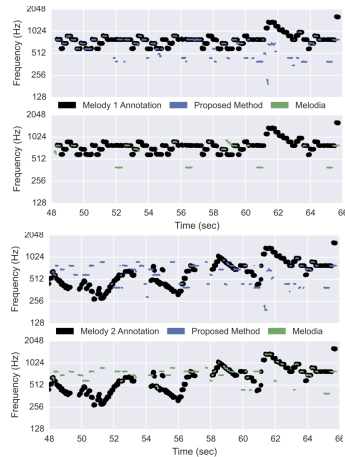


Figure 7. Outputs for melody 1 (top) and melody 2 (bottom) for a segment of “Music Delta: Latin Jazz”.

atively small training set, we would like to explore augmenting our training data through sets of time and pitch deformations.

With a slight adjustment to the evaluation metrics, our method can be easily extended to be trained on and predict melody type 3 [1] annotations, which give all feasible melody candidates at each time point, and is the most inclusive melody definition for MedleyDB. A limitation of the current method is that it assigns a single likelihood to each contour. Since the extracted contours virtually never overlap completely with the annotation, it would be desirable to be able to assign time-varying scores to each contour. To do this, we plan to explore the use of Conditional Random Fields [9] for assigning scores to contours because of their ability to incorporate temporal information. Finally, to raise the glass ceiling on performance, future work will include revisiting the contour extraction stage.

6. REFERENCES

- [1] R. M. Bittner, J. Salamon, M. Tierney, M. Mauch, C. Cannam, and J. P. Bello. MedleyDB: a Multitrack Dataset for Annotation-Intensive MIR Research. In *International Society for Music Information Retrieval Conference*, July 2014.
- [2] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [3] K. Dressler. An Auditory Streaming Approach for Melody Extraction from Polyphonic Music. In *International Society for Music Information Retrieval Conference*, 2011.
- [4] Jean-Louis Durrieu, Gaël Richard, Bertrand David, and Cédric Févotte. Source/filter model for unsupervised main melody extraction from polyphonic audio signals. *IEEE Trans. on Audio, Speech, and Language Processing*, 18(3):564–575, March 2010.
- [5] D. P. W. Ellis and G. Poliner. Classification-Based Melody Transcription. *Machine Learning Journal*, 65(2-3):439–456, December 2006.
- [6] G. D. Forney Jr. The Viterbi algorithm. *Proceedings of the IEEE*, 61(3):268–278, 1973.
- [7] M. Goto. A Real-Time Music-Scene-Description System: Predominant-F0 Estimation for Detecting Melody and Bass Lines in Real-World Audio Signals. *Speech Communication*, 43(4):311–329, September 2004.
- [8] E. Humphrey, J. P. Bello, and Y. Lecun. Feature Learning and Deep Architectures: New Directions for Music Informatics. *Journal of Intelligent Information Systems*, 41(3):461–481, December 2013.
- [9] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, 2001.
- [10] Eugene Narmour. *The Analysis and Cognition of Melodic Complexity: The Implication-Realization Model*. University of Chicago Press, November 1992.
- [11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, et al. Scikit-learn: Machine learning in python. *The Journal of Machine Learning Research*, 12:2825–2830, 2011.
- [12] C. Raffel, B. McFee, E. Humphrey, J. Salamon, O. Nieto, D. P. W. Ellis, and D. Liang. mir eval: A Transparent Implementation of Common MIR Metrics. In *International Society for Music Information Retrieval Conference*, 2014.
- [13] M. Ryyänen and A. Klapuri. Automatic Transcription of Melody, Bass Line, and Chords in Polyphonic Music. *Computer Music Journal*, 32(3):72–86, September 2008.
- [14] J. Salamon and E. Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1759–1770, Aug. 2012.
- [15] J. Salamon, E. Gómez, D. P. W. Ellis, and G. Richard. Melody Extraction From Polyphonic Music Signals: Approaches, Applications, and Challenges. *IEEE Signal Processing Magazine*, 31(2):118–134, 2014.
- [16] J. Salamon, G. Peeters, and A. Röbel. Statistical characterisation of melodic pitch contours and its application for melody extraction. In *13th Int. Soc. for Music Info. Retrieval Conf.*, pages 187–192, Porto, Portugal, Oct. 2012.
- [17] J. Salamon, B. Rocha, and E. Gómez. Musical genre classification using melody features extracted from polyphonic music signals. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 81–84, Kyoto, Japan, Mar. 2012.
- [18] H. Tachibana, T. Ono, and S. Sagayama. Melody Line Estimation in Homophonic Music Audio Signals Based on Temporal-Variability of Melodic Source. In *IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 425–428. IEEE, 2010.

COMPARISON OF THE SINGING STYLE OF TWO JINGJU SCHOOLS

Rafael Caro Repetto, Rong Gong, Nadine Kroher, Xavier Serra

Music Technology Group, Universitat Pompeu Fabra, Barcelona
{rafael.caro, rong.gong, nadine.kroher, xavier.serra}@upf.edu

ABSTRACT

Performing schools (*liupai*) in jingju (also known as Peking or Beijing opera) are one of the most important elements for the appreciation of this genre among connoisseurs. In the current paper, we study the potential of MIR techniques for supporting and enhancing musicological descriptions of the singing style of two of the most renowned jingju schools for the *dan* role-type, namely Mei and Cheng schools. To this aim, from the characteristics commonly used for describing singing style in musicological literature, we have selected those that can be studied using standard audio features. We have selected eight recordings from our jingju music research corpus and have applied current algorithms for the measurement of the selected features. Obtained results support the descriptions from musicological sources in all cases but one, and also add precision to them by providing specific measurements. Besides, our methodology suggests some characteristics not accounted for in our musicological sources. Finally, we discuss the need for engaging jingju experts in our future research and applying this approach for musicological and educational purposes as a way of better validating our methodology.

1. MOTIVATION

This paper is a joint work between an ethnomusicologist (the first author) and a team of MIR researchers in the framework of the CompMusic project. In this project we exploit jingju music characteristics (and other music traditions) with the aim of pushing forward the state of the art in MIR. In last ISMIR Conference (Taipei, 2014) jingju music received significant attention, with a specific tutorial and several papers published by members of our team [1-3], as well as the work by Tian et al. [4]. In the present paper though, the motivation has been to test the potential of current MIR methodologies to support and enhance qualitative and descriptive musicological analyses of jingju music. To this aim, we have selected one of the more relevant aspects of jingju music appreciation, which is the singing style of different performing schools;

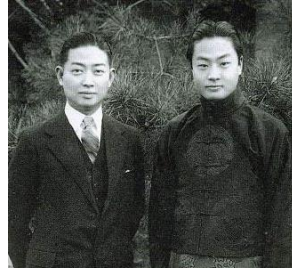


Figure 1. Mei Lanfang (left) and Cheng Yanqiu (right)

specifically, we have focused on two of the more popular ones, Mei school and Cheng school.

The paper is structured hence in the following sections. In the introduction we present the concept of jingju schools and the importance of singing style, as well as explain the purpose of the research undertaken for this paper. In the following two sections, we introduce the collection of recordings selected and the methodology proposed, and analyse the obtained results. In the discussion section we reflect on the challenges for expanding our research and present the direction of our future work. We conclude by summarising the musicological outcomes of the current research.

2. INTRODUCTION

Jingju is one of the genres of Chinese traditional theatre arts, arguably the most widespread and acclaimed one. Originally a folk art form, the actor traditionally was in charge of the whole creative process, from costumes and make-up, to acting, dancing, reciting, arranging the music and sometimes even writing (or improvising) the lyrics. In order to structure their performance, actors drew on a vast repertoire of predefined conventions handed down by tradition and which concerns every single aspect of this art. Characters of jingju plays are classified in acting categories or role-types, which define which set of conventions the actor who plays that role-type should master. The high complexity of such conventions requires the actor to specialize in the performance of just one role-type during his life-time career. Along jingju history, there were some outstanding actors that excelled in the mastery of these conventions and pushed forward the artistic standards of their respective role-types or the genre as a



© Rafael Caro Repetto, Rong Gong, Nadine Kroher, Xavier Serra.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Rafael Caro Repetto, Rong Gong, Nadine Kroher, Xavier Serra. "Comparison of the singing style of two jingju schools", 16th International Society for Music Information Retrieval Conference, 2015.

whole. Some of these masters would bring their own personalities to their performances and created personal styles. In a tradition that is orally transmitted, this would result in the appearance of *liupai*, or performing schools.¹

The first half of the 20th century was the period of major development of jingju and when the most renowned schools appeared. It saw the extraordinary development of the *dan* role-type, that portraying young or mid-aged female characters, but performed by male actors, due to social and political constrictions. Four of them gained the title of “four great *dan* actors” (*si da ming dan*),² and founded their own schools. Their strong personality, the context of market competition, and even their own physical condition caused schools of *dan* to be the ones with a greater degree of difference within one specific role-type. Among these four schools, those founded by Mei Lanfang (1894-1961) and Cheng Yanqiu (1904-1958) (Figure 1),³ respectively named Mei and Cheng schools, are the most widespread and followed ones today, currently performed in its vast majority by female actresses. These are precisely the ones we chose for our study.

Each jingju school is highly associated to a particular repertoire of plays and generally to a predominant performance skill. In fact, this repertoire is formed of plays arranged by the school founder to precisely showcase its mastery in that specific skill. In the case of Mei and Cheng schools, singing is the most representative and acclaimed aspect of their art. This aspect concerns mainly two elements, the arrangement of new tunes⁴ and the singing style. Among these two, the singing style is the feature that makes a performance instantly recognizable as belonging to any of these schools, and also one of the skills that performers, both professional and amateurs, put more effort to master. At the same time, it is one of the most important criteria for appreciating a performance among connoisseurs. The features that define singing style not only consist in the way voice is used in singing, but also in the very quality of the voice. Both of them should be considered not as natural personal qualities of particular actors, but as conventions that have to be trained and mastered. The resulting voice is to be understood hence as “an artificial voice, in the sense of displaying artifice, or art” [5], and followers of each school aim at mastering this voice quality as well.

Descriptions made in jingju musicology about singing style generally focus on its perceptual characteristics and

the psychological profile of the characters conveyed through those characteristics. Wichmann [5] quotes a typical description of Mei school’s singing style from *Zhongguo da baike quanshu* (China Great Encyclopedia) by Hu Qiaomu, in which timbre is described as “sweet, fragile, clear and crisp, round, embellished and liquid.” This timbre is considered ideal for portraying ‘natural, graceful and poised, dignified, gentle and lovely traditional women.’” In all the musicological sources consulted for this paper [5-10], description of singing style always includes this kind of terminology. However, since the aim of our research is add precision to musicological description, we have selected those characteristics for which audio features can be objectively computed. Table 1 shows the musicological characteristics selected and their corresponding audio features.

Characteristics	Audio features
Pitch register	Pitch histogram (1 st degree)
Vibrato rate variability	Vibrato rate (SD)
Volume variability	Loudness (SD)
Brightness	Spectral centroid (mean)
	LTAS
	Tristimulus
Timbre variability	Spectral flux (mean)

Table 1. Musicological characteristics and their corresponding audio features considered in our study.

In the last few years there have been several studies about singing characteristics in different Chinese traditional theatre genres, like jingju [11-12] and kunqu [13-14]. In these studies different role-types have been compared in terms of several singing characteristics by analysing monophonic recordings produced by the authors. In our work, we look in depth to one particular role-type, *dan*, and analyse singing characteristics with explicit reference to its musicological descriptions and using commercial recordings. In the following section we describe the collection of recordings and explain the methodology proposed.

3. METHODOLOGY

For this study, we have selected a collection of recordings from our jingju music research corpus [1], according to two criteria: representativeness and comparability. In order to assure that these recordings are representative of their school, we have considered both the recording artist and the recorded aria. We have looked for artists whose school filiation is explicitly stated in the release’s booklet, and arias belonging to plays for which we have literary evidence (mostly from [8]) that are representative of their school. In order to maximize comparability, we have searched for plays for which musicological literature specifically acknowledges a particular rendition from each of these two schools, as it is the case of *Su San qijie* according to [5, 8]. Since these are rare cases, due to the fact

¹ The translation of *liupai* as school can be subject to misinterpretation. Differently to other traditions, jingju schools do not imply training in specific institutions or affiliation to specific lineages. They consist in the transmission of the performance style of individual great masters, so that the reference is always the founder of the school, and not the teacher from whom the new actors or actresses learn.

² Quotes from Chinese sources are given in our translation.

³ Picture from <http://zh.wikipedia.org/wiki/程砚秋#/media/File:梅兰芳与程砚秋.JPG> (detail).

⁴ Since jingju music is created by applying pre-existing melodic conventions, it is customary to use the term arrangement (*bianqu*) instead of composition to refer to this process.

School	Work: <i>Play</i> . “Aria” (Character)	Recording: MusicBrainz ID	Length	Artist
Mei	fhc: <i>Feng huan chao</i> . “Ben ying dang su muqin Haojing ni nan” (Cheng Xue’e)	fhc - LYf: a1e4b77b-88b0-4003-b688-66e39f579dc6	7:33	Li Yufu
		fhc - SYh: 4e3b46b2-9db7-4f52-af95-e43239a6c0e1	6:56	Shi Yihong
	ssqj: <i>Su San qijie</i> . “Yu Tangchun han bei lei mang wang qian jin” (Su San)	fhc - LSS: 83d2fc7f-e1c1-4359-b417-ed9e519ecbb7	7:34	Li Shengsu
		ssqj - LSS: 067b8f25-888a-4a08-a495-cbc402846b10	7:15	
Cheng	s1n: <i>Suo lin nang</i> . “Chunqiu ting wai feng yu bao” (Xue Xiangling)	ssqj - CXqd: 87dbdf41-37ff-4f4a-83d4-7169d674579a	6:20	Chi Xiaoqiu
		s1n - CXq: 11a44af7-e29a-4c50-aa38-6139d37ca306	3:21	
	s1n - LPh: 3dcae41a-795c-4b7d-979b-1b52aa42d3a	3:06	Li Peihong	
	s1n - LGj: 1e705224-0b44-48aa-a0de-6386cda9d517	3:15	Liu Guijuan	

Table 2. Description of the recordings used in this paper. When applicable, short forms are provided.

that each school has developed its own specific repertoires, we have also searched for arias with similar music structure. This is the case of fhc and s1n, arranged in the same *shengqiang* and similar *banshi*.¹ The resulting collection of recordings is described in Table 2, together with the abbreviations used throughout the paper. We argue that the size of this collection is appropriate for the current research since we are not performing a quantitative study. Instead we are using MIR methodologies for supporting qualitative descriptions.

Figure 2 describes the methodology proposed for this paper. Each of the recordings is ripped in a lossless compressed format with a sampling rate of 44.1 kHz. We manually identify the sections containing singing voice, for which we compute the predominant melody using the vamp plug-in version of Salamon and Gómez’s algorithm [15], setting a pitch range threshold of 100 Hz to 1000 Hz, and the voicing parameter at its maximum level, 3.0. Given that a percentage of error results, pitch tracks are manually corrected (an average of 7.16% from the computed frames). In order to measure pitch register, we use the methodology proposed in [16] to compute pitch histograms and obtain the pitch of the first degree² from their peaks values. The algorithm presented in [17] is used to measure vibrato rate and extent, of which we calculate mean and standard deviation (SD). To measure the remaining features, we separate the singing voice from the accompaniment by computing a harmonic model analysis and synthesis using the methodology presented

¹ *Shengqiang* is the musical convention in jingju that determines the melodic skeleton; *banshi* refers to the metrical pattern. For more detailed information about these concepts, please refer to [1, 5].

² We use “first degree” to translate *gongyin*. This term refers to the first degree of the sung scale, and in *jianpu* notation is notated with the number 1 (http://en.wikipedia.org/wiki/Numbered_musical_notation). Although common functions are shared, we consciously avoid the term tonic, for its implications with tonality, absent in jingju music.

in [18], and apply to it standard algorithms for the computation of those features as implemented in the Essentia library [19]. For loudness we use the *Loudness* algorithm, normalizing the resulting mean to a factor of 0.5, so that SD is better comparable. To measure brightness, we compute mean and SD of the spectral centroid using the *Centroid* algorithm. In order to better understand timbre qualities, we also compute tristimulus using the *Tristimulus* algorithm, and long-term-average spectrum (LTAS) using the implementation presented in [20]. Finally, to measure timbre variability, we compute spectral flux mean and SD using the *Flux* algorithm from Essentia.

In the following section, we analyse the results obtained for each school and relate them with their corresponding musical characteristics.

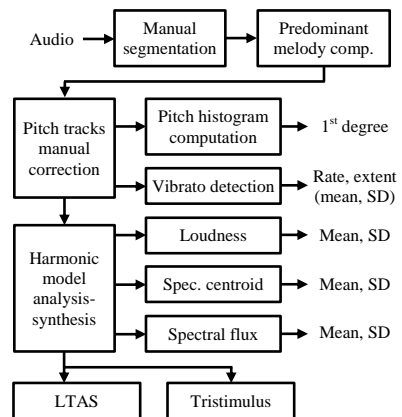


Figure 2. Block diagram of the methodology.

School	1 st deg. (Hz)	Features								
		Rate (Hz)		Vibrato Extent (cents)		Loudness SD	Spectral centroid (Hz)		Spectral flux	
		Mean	SD	Mean	SD		Mean	SD	Mean	SD
Mei	335.41	4.757	0.728	137.325	37.466	0.279	2536.739	366.968	0.121	0.063
Cheng	323.35	6.090	0.963	111.101	41.157	0.387	2136.555	451.642	0.087	0.058

Table 3. Average measurement values from each of the features computed for each school.

4. ANALYSIS OF THE RESULTS

Table 3 shows a summary with the average measurement values from each of the features computed for each school.¹ In this section we analyse how these results relate to the musicological descriptions of their corresponding musical characteristics.

According to our musicological references, pitch register in Mei is higher than in Cheng. Since pitch range of arias for the *dan* role-type is consistent across plays, approximately an octave and a major third, we take the pitch of the first degree as an indicator of pitch register. However, this degree is rarely sung in arias of this role-type. This is due to one singing convention, according to which female role-types shift their pitch register a fifth higher than male role-types, so that the modal center becomes the fifth degree. To measure the pitch of the first degree hence, we compute a pitch histogram and assign a modal degree to each peak by listening to the recordings with the aid of scores. Since we observed that the peak corresponding to the sixth degree is usually the cleanest one, we take it as reference by assigning the value of 900 cents. Figure 3 shows the resulting pitch histograms for *ssqj*-LSs and *ssqj*-CXq, compared with the equal tempered scale. It has to be noted that in jingju there is no absolute standard pitch for tuning, but it depends on the actor's or actress' needs. Notwithstanding this, a pitch is commonly assumed as reference for each role-type; for the *dan* role-type first degree is expected to be around E4 (329.63 Hz) [21]. Results in Table 3 show that this is the case for both schools, although first degree in Cheng is in average 6.28 Hz (33.30 cents) lower than E4, and Mei 5.78 Hz (30.09 cents) higher. Consequently, first degree in Cheng is in average 63.39 cents lower than Mei, more than a semitone. Results for all the recordings show that in every case first degrees from Mei are higher than those for Cheng, although the smallest difference between recordings from each school is 1.31 Hz. These results hence invite us to support the musicological description for pitch register.

Besides the aforementioned results, Figure 3 suggests that pitch histograms can shed light upon other aspects of singing style. Chen [22] has used histograms to study

that, as Figure 3 shows, compared with the equal tempered scale the fourth degree, although seldom used, is sung at a higher pitch, what is common knowledge in musicological literature. Unexpectedly, the higher octave of the first degree appears in the histograms slightly shifted higher, especially in Mei school, for which we have not found literary evidence. Besides, peak shape differences, cleaner and with lower valleys for Cheng, also suggest differences in singing style, probably regarding vibrato and ornamentation. These observations invite us to argue that pitch histograms could be further exploited for the characterisation of singing style and explore features unnoticed or not explicitly accounted for in our references.²

According to the sources consulted, Cheng “excels in

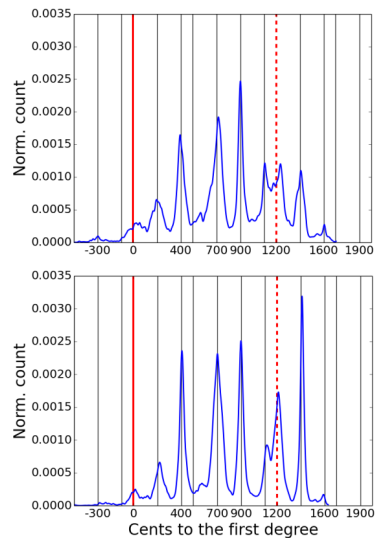


Figure 3. Pitch histograms for *ssqj*-LSs (top) and *ssqj*-CXq (bottom). Vertical lines show the equal tempered scale, solid red line marks the first degree, and dotted red line marks its higher octave.

¹ Detailed results and more plots can be found in <https://github.com/jingjuschools/jingjuschoolsISMIR2015>

² Differences in peaks height indicate different melodic preferences in each school, what concerns tune arrangement, an issue not considered in this paper.

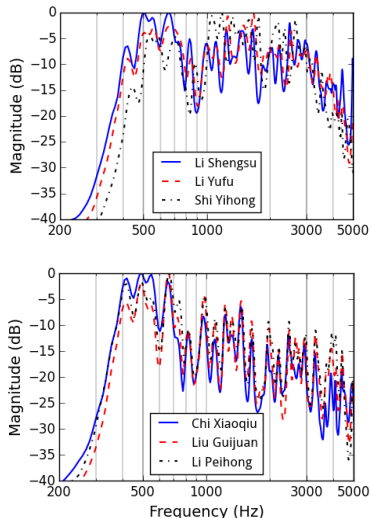


Figure 4. LTAS for the three recordings of fhc (top) and the three recordings of s1n (bottom).

using slow and fast vibratos” [9]. Consequently, the feature that can better reflect this characteristic is the standard deviation of vibrato rate. As can be seen in Table 3, this value is higher in Cheng than in Mei, and this is also the case for every recording. However, the difference is less than 0.3 Hz, what is barely appreciable by human ear. Variability in vibrato extent, as reflected in our results, is slightly higher in Cheng than in Mei, but the difference in this case is even less significant. Besides, specific results from each recording are less consistent, since some instances from Mei school show higher SD in vibrato extent than others from Cheng, and vice versa. What our results clearly show though, is that vibrato in Mei is considerably slower and wider than in Cheng, a feature that is consistent across all the recordings. Interestingly enough, we have not found such a remark in our musicological sources.

Results obtained for loudness variance also support musicological description, which takes volume variability as a characteristic of Cheng school compared to Mei. Results for each recording are less consistent than for other features, finding one case in Mei with higher SD than the lowest value for a recording in Cheng. These results, however, ought to be taken carefully. Firstly, they might have been affected by possibly different mixing levels in the production process. Secondly, being loudness a perceptual feature, the algorithm used is an approximation to it by a simple modification of amplitude values, what prevent us to take them as a faithful representation of the characteristic measured.

Our musicological references agree that timbre in Mei school is brighter than in Cheng. To measure brightness, we have computed spectral centroid mean and SD. Values for the mean effectively show that Mei has brighter timbre than Cheng, as an average and for every recording in each school. Given the complexity of timbre, we have also looked at LTAS, a feature that has been used in [23] to study and compare vocal tract and formant structure. Figure 4 shows the LTAS for the three performances of fhc and s1n. To focus on the region with greater loudness, plots show the region between 200 Hz and 5000 Hz, with a logarithmic scale in the x-axis. In these plots it can be observed how frequencies over 1000 Hz are considerably higher in Mei than in Cheng, as well as the peaks with the highest loudness, contributing thus to timbre brightness. Besides this information, LTAS allows us to compare vocal tract between performers in each school, so that we can observe that timbre similarity is higher in Cheng than in Mei. This method hence seems promising in order to characterise individual actors’ or actresses’ particularities within the overall requirements of the school.

Tristimulus has also been used to study and compare timbre qualities [24]. Figure 5 shows that in average both the second and the third of the three output components measured by tristimulus have higher values for Mei than for Cheng, what once more support the higher weight of higher partials in Mei. This figure also suggests a promising tool for classification according to timbre quality.

Finally, results for spectral flux, computed with the aim of measuring timbre variability, show a greater value in Mei than in Cheng, a difference which is consistent across all the recordings, with special homogeneity in Mei. Literature however remarks Cheng’s timbre variability as a defining trait of this school. It is also interesting to notice that the SD value for spectral centroid also shows a higher value in Cheng, although is not as consistent as the spectral flux values across all the recordings. Since this descriptor was computed only for the sections of the recordings that contained singing, we re-

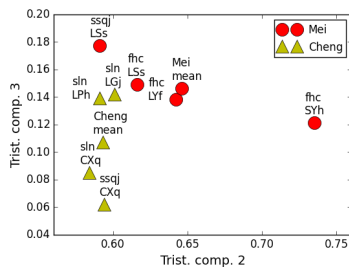


Figure 5. Scatter plot displaying values for the 2nd and 3rd tristimulus components for each recording and the mean for each school.

computed it setting different loudness thresholds for the transition frames in order to discard an influence of the segmentation. Results didn't change any tendency towards a higher value in Cheng than in Mei. How to approach timbre variability in jingju singing using audio analysis features remains hence an issue for future research, as discussed in the following section.

5. DISCUSSION

The analysis of the results presented in the previous section suggests that the methodology proposed in this paper is promising for the intended task, namely, supporting musicological description using audio analysis features. However, there are also some challenges that have to be addressed when extending this research in future work.

We aim to improve our methodology in the following two senses: automatizing most of the steps for audio analysis and improving existing algorithms to better fit jingju music characteristics. Some researchers in our team are currently working on improving the automatic segmentation method by Chen [22], adapting Ishwars' methodology [25] to jingju music for better extraction of predominant melody, and developing new algorithms for automatic computation of the first degree. Improvement of the harmonic model analysis and synthesis as presented in [18] is also to be undertaken in the near future.

Arguably, the bigger challenge for the continuation of this research is gaining the engagement of jingju musicologists. In this paper we have aimed to show how the present approach can benefit musicological work. Yet to that aim we have consciously avoided most of the terminology that is more commonly used by experts when describing singing style. The authors have not yet agreed on a methodology from an MIR point of view for approaching descriptions such as "sweet" (*tian*), "mellow" (*run*), "fragile" (*cui*), "round" (*yuan*), or "wide" (*kuan*). Even when considering a characteristic like timbre variability, whose study by means of spectral flux disagrees with musicological descriptions, we wonder how much of this disagreement is due to difficulties in establishing a common terminology between these two disciplines. From the field of MIR there have been recent calls for a better understanding of the musical content of commonly computed descriptors [26]. The authors argue that collaborative research as the one undertaken here would also encourage jingju experts to reflect on their terminology in terms of audio analysis features, and hopefully would gain complementary precision for those concepts.

Besides supporting musicological research, the use of audio features for qualitative analysis can be exploited for educational purposes. Jingju is a tradition that relies on oral transmission for training young actors and actresses. The key method in this training tradition consists on "teaching by mouth and heart" (*kou chuan xin shou*), that is, the teacher sings and the student repeats as many times as needed for achieving an acceptable standard. Recently,

new technologies are being used as part of this process. Students use their cell phones to record their teachers, and audio and video recordings of performances are easily accessible in the web. Technologies that could automatically evaluate the degree of similarity between the teacher's and the student's performance, and moreover offer a precise description of dissimilarities, would guide the trainee in better understanding his or her own learning process. The aim of such technologies would be performing qualitative analysis of audio recordings, similar to the ones implemented in this paper. Building upon the results obtained and the methodology tested in the current work, we have started to develop such educational tools. To this goal we will require closer collaboration with jingju experts. The involvement of these experts and the acceptance of the educational tools by jingju trainees will also provide better evaluation methods for our research.

6. CONCLUSIONS

The current paper has studied the potential of using audio analysis features for supporting and enhancing the musicological description of singing style in two jingju schools for the *dan* role-type, namely Mei and Cheng. Our results support the description given in musicological literature for most of the characteristics analysed, and add precision to them. Pitch register in Cheng school is in average 63.39 cents lower than Mei. Variability in vibrato rate is slightly higher in Cheng, what agrees with musicological description, but less than 0.3 Hz. Volume variability as a characteristic of Cheng school has been supported by the measure of loudness, whose SD is 38.71% higher in this school than in Mei. That this school is brighter in timbre than Cheng is supported by the mean value of its spectral centroid, 18.73% higher, but also by measurements in LTAS and trisstimulus. Besides supporting these descriptions, our method also suggests some characteristics not explicitly accounted for in the sources consulted: vibrato in Mei is in average 1.33 Hz slower and 26.22 cents wider than in Cheng, pitch histograms suggest new characterisations for tuning and intonation, and LTAS looks promising for comparing vocal tracts of singers within a school. Only in the case of spectral flux, computed for studying timbre variability, the results do not support the musicological description, an issue that will be addressed in future research. In the light of these results, we have started to extend this methodology for the development of educational tools, a project in which we hope to gain the engagement of jingju experts, who could benefit from this approach for their own research.

7. ACKNOWLEDGEMENTS

This research is funded by the European Research Council under the European Union's Seventh Framework Program, as part of the CompMusic project (ERC grant agreement 267583).

8. REFERENCES

- [1] R. Caro Repetto, and X. Serra: "Creating a Corpus of Jingju (Beijing Opera) Music and Possibilities for Melodic Analysis," *ISMIR 2014*, pp. 313-318, 2014.
- [2] A. Srinivasamurthy, R. Caro Repetto, H. Sundar, and X. Serra: "Transcription and Recognition of Syllable based Percussion Patterns: The Case of Beijing Opera," *ISMIR 2014*, pp. 431-436, 2014.
- [3] S. Zhang, R. Caro Repetto, and X. Serra: "Study of the Similarity between Linguistic Tones and Melodic Pitch Contours in Beijing Opera Singing," *ISMIR 2014*, pp. 343-348, 2014.
- [4] M. Tian, G. Fazekas, D. Black, and M. Sandler: "Design and Evaluation of Onset Detectors Using Different Fusion Policies," *ISMIR 2014*, pp. 631-636.
- [5] E. Wichmann: *Listening to Theatre: The Aural Dimension of Beijing Opera*, University of Hawaii Press, Honolulu, 1991.
- [6] H. Li 李海涓: "Jingju qingyi 'Mei' 'Cheng' er pai zai changfa shang de tong yu yi" 京剧青衣“梅”“程”二派在唱法上的同与异 (Similarities and differences in the singing techniques between the two schools of jingju qingyi Mei and Cheng), *Zhejiang yishu zhiye xueyuan xuebao*, Vol. 11, No. 1, pp. 50-55, 2013.
- [7] R. Wang 汪入立: "Mei pai changqiang yinyue de meixue ping" 梅派唱腔音乐的美学品格 (Character of music aesthetics in Mei school's singing), *Yishu bai jia*, 1996, No. 1, pp. 40-47.
- [8] T. Wu 吴同宾, and Y. Zhou 周亚勋: *Jingju zhishi cidian* 京剧知识词典 (Dictionary of jingju knowledge), Tianjin renmin chubanshe, Tianjin, 2006.
- [9] S. Yu 俞淑华: "Chuyi Chengpai de changqiang yu banzou" 刍议程派的唱腔与伴奏 (My humble opinion about Cheng school's singing and instrumental accompaniment), *Zuojia zazhi*, 2012, No. 1, pp. 209-210.
- [10] S. Yu 俞淑华: "Lun jingju 'si da ming dan' de changqiang yinse" 论京剧“四大名旦”的唱腔音色 (Discussing singing timbre in jingju's 'four great dan actors'), *Ming zuo xinshang*, 2011, No. 33, pp. 114-115.
- [11] J. Sundberg, L. Gu, Q. Huang, and P. Huang: "Acoustical study of classical Peking Opera singing," *Journal of Voice*, Vol. 26, No. 2, pp. 137-143, 2012.
- [12] L. Yang, M. Tian, and E. Chew: "Vibrato characteristics and frequency histogram envelopes in Beijing opera singing," *5th International Workshop on Folk Music Analysis*, pp. 139-140, 2015.
- [13] L. Dong, J. Sundberg, and J. Kong: "Loudness and Pitch of Kunqu Opera," *Journal of Voice*, Vol. 28, No. 1, pp. 14-19, 2014.
- [14] L. Dong, J. Kong, and J. Sundberg: "Long-term-average spectrum characteristics of Kunqu Opera singers' speaking, singing and stage speech," *Logopedics Phoniatrics Vocology*, Vol. 39, No. 2, pp. 72-80, 2014.
- [15] J. Salamon, and E. Gómez: "Melody Extraction From Polyphonic Music Signals Using Pitch Contour Characteristics," *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 20, No. 6, pp. 1759-1770, 2012.
- [16] G. K. Koduri, V. Ishwar, J. Serrà, X. Serra, and H. Murthy: "Intonation analysis of ragas in Carnatic music," *JNMR*, Vol. 43, No. 1, pp. 72-93.
- [17] P. Herrera, and J. Bonada: "Vibrato Extraction and Parameterization in the Spectral Modeling Synthesis Framework," *DAFx*, 1998.
- [18] X. Serra, and J. Smith: "Spectral Modeling Synthesis: A Sound Analysis/Synthesis Based on a Deterministic plus Stochastic Decomposition," *Computer Music Journal*, Vol. 14, No. 4, pp. 12-24, 1990.
- [19] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra: "ESSENTIA: an Audio Analysis Library for Music Information Retrieval," *ISMIR 2013*, pp. 423-498, 2013.
- [20] T. Kinnunen, V. Hautamäki, and P. Fränti: "On the Use of Long-Term Average Spectrum in Automatic Speaker Recognition," *Proceedings of the 5th International Symposium on Chinese Spoken Language Processing*, pp. 559-567, 2006.
- [21] B. Cao 曹宝荣: *Jingju changqiang banshi jiedu: Xia ce* 京剧唱腔板式解读·下册 (Deciphering banshi in jingju singing: Second volume), Renmin yinyue chubanshe, Beijing, 2010.
- [22] K. Chen: *Characterization of Pitch Intonation of Beijing Opera*, Master thesis, Universitat Pompeu Fabra, Barcelona, 2013.
- [23] J. Sundberg: *The Science of the Singing Voice*, Northern Illinois University Press, Dekalb, 1987.
- [24] M. Campbell, and C. Greated: *The Musician's Guide to Acoustics*, Oxford University Press, Oxford, 1987.
- [25] V. Ishwar: *Pitch Estimation of the Predominant Vocal Melody from Heterophonic Music Audio Recordings*, Master Thesis, Universitat Pompeu Fabra, Barcelona, 2014.
- [26] B. Sturm: "A Simple Method to Determine if a Music Information Retrieval System is a 'Horse'," *IEEE Transactions on Multimedia*, Vol. 16, No. 6, pp. 1636-1644, 2014.

Oral Session 4

Mixed

IMPROVING OPTICAL MUSIC RECOGNITION BY COMBINING OUTPUTS FROM MULTIPLE SOURCES

Victor Padilla
Lancaster University
victor.padilla.
mc@gmail.com

Alex McLean
University of Leeds
a.mclean@
leeds.ac.uk

Alan Marsden
Lancaster University
a.marsden@
lancaster.ac.uk

Kia Ng
University of Leeds
k.c.ng@
leeds.ac.uk

ABSTRACT

Current software for Optical Music Recognition (OMR) produces outputs with too many errors that render it an unrealistic option for the production of a large corpus of symbolic music files. In this paper, we propose a system which applies image pre-processing techniques to scans of scores and combines the outputs of different commercial OMR programs when applied to images of different scores of the same piece of music. As a result of this procedure, the combined output has around 50% fewer errors when compared to the output of any one OMR program. Image pre-processing splits scores into separate movements and sections and removes ossia staves which confuse OMR software. Post-processing aligns the outputs from different OMR programs and from different sources, rejecting outputs with the most errors and using majority voting to determine the likely correct details. Our software produces output in MusicXML, concentrating on accurate pitch and rhythm and ignoring grace notes. Results of tests on the six string quartets by Mozart dedicated to Joseph Haydn and the first six piano sonatas by Mozart are presented, showing an average recognition rate of around 95%.

1. INTRODUCTION

Musical research increasingly depends on large quantities of data amenable to computational processing. In comparison to audio and images, the quantities of symbolic data that are easily available are relatively small. Millions of audio recordings are available from various sources (often at a price) and images of tens of thousands of scores are freely available (subject to differences in copyright laws) in the on-line Petrucci Music Library (also known as IMSLP). In the case of data in formats such as MEI, MusicXML, Lilypond, Humdrum kern, Musedata, and even MIDI, which give explicit information about the notes that make up a piece of music, the available quantities are relatively small. The KernScores archive [21] claims to contain 108,703 files, but many of these are not complete pieces of music. Mutopia, an archive of scores

in the Lilypond format, claims to contain 1904 pieces though some of these also are not full pieces. The Musescore collection of scores in MusicXML gives no figures of its contents, but it is not clearly organised and cursory browsing shows that a significant proportion of the material is not useful for musical scholarship. MIDI data is available in larger quantities but usually of uncertain provenance and reliability.

The creation of accurate files in symbolic formats such as MusicXML [11] is time-consuming (though we have not been able to find any firm data on how time-consuming). One potential solution to this is to use Optical Music Recognition (OMR) software to generate symbolic data such as MusicXML from score images. Indeed, Sapp [21] reports that this technique was used to generate some of the data in the KernScores dataset, and others have also reported on the use of OMR in generating large datasets [2, 3, 6, 23]. However, the error rate in OMR is still too high. Although for some MIR tasks the error rate may be sufficiently low to produce usable data [2, 3, 23], the degree of accuracy is unreliable.

This paper reports the results of a project to investigate improving OMR by (i) image pre-processing of scanned scores, and (ii) using multiple sources of information. We use both multiple recognisers (i.e., different OMR programs) and multiple scores of the same piece of music. Preliminary results from an earlier stage of this project were reported in [16]. Since then we have added image pre-processing steps, further developments of the output-combination processes, and mechanisms for handling piano and multi-part music. We also report here the results of much more extensive testing. The basic idea of combining output from different OMR programs has been proposed before [4, 5, 13] but this paper presents the first extensive testing of the idea, and adds to that the combination of outputs from different sources for the same piece of music (different editions, parts and scores, etc.).

In view of our objective of facilitating the production of large collections of symbolic music data, our system batch processes the inputted scores without intervention from the user. The basic workflow is illustrated in Figure 1. Each step of the process is described in subsequent sections of this paper, followed by the results of a study that tested the accuracy of the process.

Music notation contains many different kinds of information, ranging from tempo indications to expression



© Victor Padilla, Alex McLean, Alan Marsden & Kia Ng.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Victor Padilla, Alex McLean, Alan Marsden & Kia Ng. "Improving Optical Music Recognition by Combining Outputs from Multiple Sources", 16th International Society for Music Information Retrieval Conference, 2015.

markings, and to individual notes. The representation varies in both score and symbolic music data formats. In this study we assume the most important information in a score to be the pitch and duration of the notes. Therefore, we have concentrated on improving the accuracy of recognition of these features alone. Grace notes, dynamics, articulation, the arrangement of notes in voices, and other expression markings, are all ignored. However, when a piece of music has distinct parts for different instruments (e.g., a piece of chamber music) we do pay attention to those distinct parts. For our purposes, a piece of music therefore consists of a collection of “parts”, each of which is a “bag of notes”, with each note having a “pitch”, “onset time” and “duration”.

Broadly speaking, we have been able to halve the number of errors made by OMR software in recognition of pitches and rhythms. However, the error rate remains relatively high, and is strongly dependent on the nature of the music being recognised and the features of the inputted score image. We have not tested how much time is required to manually correct the remaining errors.

2. BACKGROUND

2.1 Available Score Images

In the past it was common for research projects to scan scores directly. For example, in [2, 3], pages were scanned from the well-known jazz ‘Fake Book’. Now, however, many collections of scans are available on-line. The largest collection is the Petrucci Music Library (also called IMSLP),¹ which in April 2015 claimed to contain 313,229 scores of 92,019 works. Some libraries are placing scans of some of their collections on-line and some scholarly editions, such as the Neue Mozart Ausgabe (NMA),² are also available on-line. Scores available on-line are usually of music which is no longer in copyright, and date from before the early twentieth century.

Most of these scans are in PDF format and many are in binary images (one-bit pixels). Resolution and qualities of the scans varies.

2.2 OMR Software

Eight systems are listed in a recent survey as ‘the most relevant OMR software and programs’ [18]. Of these we found four to be usable for our purpose: Capella-Scan 8.0, SharpEye 2.68, SmartScore X2 Pro and PhotoScore Ultimate 7.³ All four pieces of software produce output in

MusicXML format [11]. They differ in the image formats which they take as input, and also in whether they can take multiple pages as input. The lowest common denominator for input is single-page images in TIFF format.

Although our objective was not to evaluate the different OMR programs, we did find that the programs differed considerably in their accuracy when applied to different music. No one OMR program was consistently better than the rest. An indication of the differences between them is given in the results section below.

3. OUR MULTIPLE-OMR SYSTEM FOR IMPROVED ACCURACY

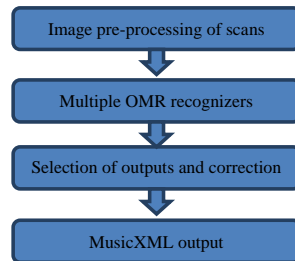


Figure 1. Basic workflow of the proposed system.

3.1 Image Pre-Processing

As stated above, the common required input for the OMR programs is single-page TIFF images. The first steps in the image pre-processing are therefore to split multiple-page scores into individual pages, and to convert from PDF, which is the most common format used for downloadable score images, including those from IMSLP.

The other pre-processing steps depend on the detection of staves in the image. In general we use the Miyao [14] staff finding method as implemented in the Gamera software,⁴ which locates equally spaced candidate points and links them using dynamic programming. This method did not perform well at detecting short ossia staves. For this we applied the Dalitz method (in class `StaffFinder_dalitz`) from the same library [9].⁵

Further processing is required to recognise systems in the images. Contours are detected using the `findContours` function [22] of the OpenCV library for computer vision,⁶ with those containing staves marked as systems. Each of the remaining contours are then assigned to the nearest system, looking for the largest bounding box overlap, or simply the nearest system on the y-axis.

None of the OMR programs used handled divisions between movements properly: the music in an image was always assumed by the software to be a single continuous

¹ www.imslp.org

² dme.mozarteum.at

³ www.capella.de, www.visiv.co.uk, www.musitek.com, www.sibelius.com/products/photoscore The other four listed by Rebelo et al. are ScoreMaker (cmusic.kawai.jp/products/sm), which we found to be available only in Japanese, Vivaldi Scan, which appears to have been withdrawn from sale, Audiveris (audiveris.kenai.com), open-source software which we found to be insufficiently robust (version 5 was under development at the time of this project), and Gamera (gamera.informatik.hs-nr.de), which is not actually an OMR system but instead a toolkit for image processing and recognition.

⁴ gamera.informatik.hs-nr.de

⁵ music-staves.sf.net

⁶ opencv.org

piece of music. This is not problematic in a process which depends on the intervention of a human operator. However, this is inefficient and not scalable for large-scale batch processing. Therefore, we implemented a process which recognises the beginnings of movements, or new sections, from the indentation of the first system of staves. Where indented staves are detected, the output is two or more TIFF files containing images of those staves which belong to the same movement or section. This procedure correctly separated all cases in our test dataset.

A second common source of error was found to be 'ossia' segments. An ossia is a small staff in a score, generally placed above the main staff, that offers an alternative way of playing a segment of music, for example, giving a possible way of realising ornaments. The OMR programs tended to treat these as regular staves, leading to significant propagation errors. Since, as indicated above, our aim was to improve the recognition accuracy of pitches and rhythms only, ossia staves would not contain useful information consistent with this aim. The best course of action was to simply remove them from the images. Therefore, the minimum bounding rectangle which included any staff that was both shorter than the main staff and smaller in vertical size, and any symbols attached to that staff (in the sense of there being some line of black pixels connected to that staff), was removed from the image. We did not separately test the ossia-removal step, but found that a few cases were not removed properly.

3.2 Post-Processing: Comparison and Selection

The images resulting from the pre-processing steps described above are then given as input to each of the four OMR programs. Output in MusicXML from each OMR program for every separate page for each score is combined to create a single MusicXML file for that score and that OMR program. As mentioned above, we chose to concentrate on the most important aspects of musical information, and so the post-processing steps described below ignored all grace notes and all elements of the MusicXML which did not simply describe pitch and rhythm. Most of the processing is done using music21 [8], using its data structures rather than directly processing the MusicXML.

The most common errors in the output are incorrect rhythms and missing notes. Occasionally there are errors of pitch, most often resulting from a failure to correctly recognise an accidental. Rests are also often incorrectly recognised, leading to erroneous rhythms. Sometimes larger errors occur, such as the failure to recognise an entire staff (occasionally a result of curvature in the scan), and the failure to correctly recognise a clef, can lead to a large-scale propagation of a single error.

The aim of our post-processing of MusicXML outputs was to arrive at a single combined MusicXML output which contained a minimum number of errors. However, this aim was challenging to fulfil because it was not pos-

sible to determine, based on the MusicXML alone which details are correct and which are incorrect. Our general approach, following Bugge *et al.* [4] is to align the outputs and to use majority voting as a basis for deciding which details are correct and which are incorrect.

The first post-processing steps apply to music with more than one voice on a single staff (as is common in keyboard music). Different OMR programs organise their output in different ways and some reorganisation is necessary to ensure that proper matching can take place. The steps in this part of the process are:

- a) **Filling gaps with rests.** In many cases, rests in voices are not written explicitly in the score, and the OMR software recognises rests poorly. Furthermore, while music21 correctly records the timing offsets for notes in voices with implied rests, MusicXML output produced from music21 in such cases can contain errors where the offset is ignored. To avoid these problems, we fill all gaps or implied rests with explicit rests so that all voices in the MusicXML contain symbols to fill the duration from the preceding barline.
- b) **Converting voices to chords.** The same music can be written using chords in some editions but separate voices in others. (See Figure 2 for an example.) To allow proper comparison between OMR outputs, we convert representations using separate voices into representations that use chords.



Figure 2. Extracts from the NMA and Peters editions of Mozart piano sonata K. 282, showing chords in the NMA where the Peters edition has separate voices.

- c) **Triplets.** In many piano scores, triplets are common, but not always specified. Some OMR programs correctly recognise the notes, but not the rhythm. Our application detects whether the length of a bar (measure) matches with the time signature in order to determine whether triplets need to be inserted where notes beamed in threes are detected.

A grossly inaccurate output can lead to poor alignment and poor results when combining outputs. Therefore, it is better to exclude outputs which contain a lot of errors from subsequent alignment and voting, but again it is not possible to determine whether an output is grossly inaccurate on the basis of its contents alone. We once again

employed the idea of majority voting: an output which is unlike the others is unlikely to be correct.

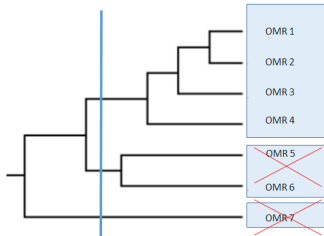


Figure 3. Example of phylogenetic tree with pruning of distant branches.

To determine how much outputs are like each other, we adopted the mechanism of ‘phylogenetic trees’ by UPGMA [24]. This mechanism is designed to cluster DNA sequences according to their similarity. Instead of a DNA string, each staff can be converted into a pitch-rhythm sequence and compared in pairs using the Needleman-Wunsch algorithm [15]. This process leads to the establishment of a similarity matrix and phylogenetic tree. Once the tree is configured, distant branches can be removed by a straightforward operation. So far, our best results have been obtained by using the three or four closest OMR outputs and discarding the rest. An evaluation of a more complex algorithm for pruning trees is left for future research.

3.3 Post-Processing: Alignment and Error-Correction

In order to determine the correct pitch or duration of a note, on the basis of majority voting, we need to know that the possible values in the different outputs refer to the same note. Therefore, it is necessary to align the outputs so that: the appropriate details in each output can be compared; a majority vote can be made; the presumably correct details can be selected; and a composite output can be constructed using these details.

There are two methods to align outputs, which we refer to as ‘top-down’ and ‘bottom-up’. The first method aims to align the bars (measures) of the outputs so that similar bars are aligned. The second method aims to align the individual notes of the outputs irrespective of barlines. The first works better in cases where most barlines have been correctly recognised by the OMR software but the rhythms within the bars might be incorrectly recognised due to missing notes or erroneous durations. The second works better in cases where most note durations have been correctly recognised but the output is missing or contains extra barlines. In the following section, we explain the bottom-up process and how we combine the outputs from parts in order to obtain a full score. An explanation of our top-down approach used in earlier work can be found in [16].

3.3.1 Bottom-Up Alignment and Correction, Single Parts

Bottom-up alignment is applied to sequences of symbols from a single staff or a sequence of staves that correspond to a single part in the music. This might come from MusicXML output of OMR applied to an image of a single part in chamber music, such as a string quartet, full score, or keyboard music in which the staves for the right and left hands are separated. Each non-ignored symbol represented in the MusicXML (time signature, key signature, note, rest, barline, etc.) is converted to an array of values to give the essential information about the symbol. The first value gives the type of symbol plus, where appropriate, its pitch and/or duration class (i.e., according to the note or rest value, not the actual duration taking into account triplet or other tuplet indications). Alignment of OMR outputs, once again using the Needleman-Wunsch algorithm, is done using these first values only. In this way we are able to avoid alignment problems which might otherwise have occurred from one edition of a piece of music indicating triplets explicitly and another implicitly implying triplets, or from one OMR recognising the triplet symbol and another not recognising the symbol. All outputs are aligned using the neighboring algorithm [20], starting with the most similar pair. A composite output is generated which consists of the sequence of symbols which are found to be present at each point in at least half of the outputs.

Figure 4. Example of removal of voices for alignment and subsequent reconstruction.

In the case of music where a staff contains more than one voice, such as in piano music, we adopt a procedure which creates a single sequence of symbols (see Figure 4). To achieve this, notes and rests are ordered first by their onset time. Next, rests are listed before notes and higher notes listed before lower notes. The result is a canonical ordering of symbols that make up a bar of multi-voice music. This means that identical bars will always align perfectly. Voice information (i.e., which voice a note belongs to) is recorded as one of the values of a note’s array. However, this information is not taken into account when the correct values are determined by majority voting. Instead, voices are reconstructed when the aligned outputs are combined. Although, this means that the allocation of notes to voices in the combined output

might not match any of the inputs (and indeed might not be deemed correct by a human expert), we found considerable variability in this aspect of OMR output and therefore could not rely upon it. At the same time, as in the pre-processing of MusicXML outputs from the OMR programs, additional rests are inserted into the combined output in order to ensure that every voice is complete from the beginning of each bar.

3.3.2 Top-Down Combined Output, Full Score

To generate a MusicXML representation of the full score, the results of alignment of the separate parts/staves need to be combined. Often the results for the constituent parts do not contain the same number of bars (measures), usually because of poor recognition of ‘multi-rests’ (i.e., several bars of rest that look like a single bar), or because of missing barlines. Sometimes OMR software inserts barlines where there are none. To achieve parts with the same number of bars and the best construction of the full score, the following steps are implemented:

a) **Finding the best full score OMR.** As a result of errors in recognising systems and staves, OMR is more likely to increase than reduce the number of bars. In the case of chamber music (e.g., string quartets), it is common to find OMR putting bars from one part in another part, adding extra rest bars and increasing the global number of bars. A simple algorithm is therefore used to select the OMR output that contains the smallest number of bars and the correct number of parts. We have found this to work correctly in most cases. As an example, the 724 bars in Mozart’s string quartet K. 387 can be converted into 747, 843, 730 and 764 bars by different OMR programs. Figure 5 shows the result of OMR errors in interpreting the arrangement of staves, in this case causing two bars to be converted into four. One system of staves is mis-interpreted as two staves, and what is actually the second violin part is read as a continuation of the first violin part. In this case there is also the very common error of misreading the alto clef.



Figure 5. Displacement of parts in a string quartet.

b) **Aligning parts against the best full score.** Every bar of each part is converted into a sequence of hash values on the basis of the pitch and rhythm of the contents of the bar. The parts are aligned bar-by-bar (top-down approach) using the Needleman-Wunsch algorithm, and empty bars are introduced where needed. To determine

the similarity of each pair of bars, the Needleman-Wunsch algorithm is used to align the contents of the two bars, and the aggregate cost of the best alignment taken as a measure of the similarity of the bars. For further detail, see [16]. This procedure results in correct vertical alignment of most of the bars and adds multi-rests that were not properly recognised in single parts.

3.4 Implementation

Our research was conducted using the Microsoft Windows operating system. This was because SharpEye only operates with this system. (The other three have versions for Windows and Macintosh systems.) Software was written in Python and made use of the Gamera and music21 libraries.

The items of OMR software used were designed for interactive use, so it was not a simple matter to integrate them into a single workflow. For this purpose we used the Sikuli scripting language.¹

A cluster of six virtual machines running Windows Server 2012, controlled by a remote desktop connection, was setup to run the OMR software and our own pre- and post-processing software. To generate MusicXML from a set of scans of a piece of music, the required scans need to be stored in a particular directory shared between the different machines. The setup also provides different levels of Excel files for evaluating results.

With the exception of the commercial OMR programs, the software and documentation are available at <https://code.soundsoftware.ac.uk/projects/multiomr> and at <http://github.com/MultiOMR>.

4. EVALUATION

4.1 Materials

To test our system, we chose to use string quartets and piano sonatas by Mozart, because both scans and pre-existing symbolic music data for these pieces are available. The pieces tested were the string quartets dedicated to Joseph Haydn (K. 387, 421, 428, 458, 464 and 465) and the first six piano sonatas (K. 279, 280, 281, 282, 283 and 284). The sources have been taken from IMLSP (Peters edition, full scores and parts) and NMA (full scores). Ground truth files in Humdrum Kern format or MusicXML (when available) were downloaded from KernScores.² Two movements, (K. 428, mov. 4 and K. 464, mov. 1) are not yet complete on KernScores and so have not been evaluated. The string-quartet dataset included a total of 459 pages of music notation and the piano-sonata set 165 pages.

4.2 Results

For each piece, the output resulting from our system was compared with the data derived from KernScores and the

¹ www.sikuli.org

² kern.ccarh.org

recognition rate for notes was calculated (i.e., the percentage of notes in the original which were correctly represented in the output). Errors found in the output of our system compared to the ground truth were recorded in Excel and MusicXML files. These provided information about each incorrect note and its position in the score, the accuracy of the overall result, and the accuracy of each OMR program, plus colour-coding in the MusicXML file to indicate omissions, errors and insertions.

	OUT	CP	PS	SE	SS
Piano	95,11	74,26	86,13	91,86	85,40
String Quartet	96,12	47,84	81,47	86,65	82,40
Average	95,61	61,05	83,80	89,25	83,90

Table 1. Overall recognition rates. OUT = our system; CP = Capella-Scan; PS = PhotoScore; SE = SharpEye; SS = SmartScore.

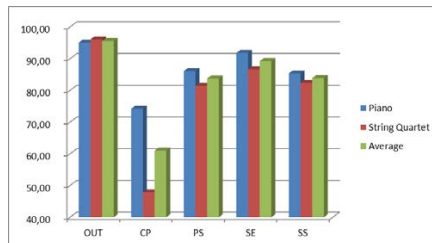


Figure 8. Overall recognition rates. For codes see the caption to Table 1.

Summary results are shown in Figure 8 and Table 1. As can be seen, overall, the output of our system was found to be better than all the OMR programs. The difference in recognition rates is higher for the string quartets due to very poor recognition in some cases for the Peters full score edition. One of the main strengths of our system rests with having consistent recognition rates of around 95% in most cases. This can be attributed to the automatic removal of scores poorly recognised at the phylogenetic-tree stage. This pruning removes a large amount of noise introduced by some OMR systems. A second strength is the lack of hard-coded rules. For instance, some OMR programs are better at recognising triplets and others at detecting pitch. Furthermore, the situation can change with new versions of OMR software and the introduction of new OMR programs. For our system to incorporate new versions and new OMR programs, all that is required is to add the necessary scripts in Sikuli to use those programs to generate MusicXML output from a given set of score images.

The entire process (six complete string quartets using parts and full scores), employing six virtualised machines in parallel, with an Intel Xeon processor of 3.30GHz, takes around 3 hours to complete.

5. FUTURE PROSPECTS

We have shown that by using pre-processing techniques and combining results from different OMR programs and different sources, we can significantly reduce the number of errors in optical music recognition. For those compiling corpuses of musical data in symbolic format, this would increase efficiency and save considerable effort. Furthermore, the reduction in error rate means that research which relies on large quantities of musical data which was previously impossible because it would be too costly to compile the data might now become possible. Large quantities of data can now be easily derived from scans available from IMSLP and elsewhere. Although errors remain in the output, the reduced error rate compared with raw OMR programs output will enable more valid results to be derived from statistical studies which previously lacked validity. Nevertheless, it should be noted that the number of errors which remain will still be too high for musicological research which assumes near 100% accuracy in the data.

Our research has made clear the limitations of current commercial OMR software. For example, in [16] we proposed the image pre-processing to extract separate bars in order to arrive at a better alignment of outputs that come from different OMR programs. However, we discovered that the OMR programs were generally incapable of producing any useful output from a single bar of music so we had to abandon this idea. It is our judgement, based on the project reported here, that further improvement of our system will require the limitations associated with current OMR software to be overcome. This may require a fundamentally different approach to currently available OMR programs.

Some possible directions have been proposed in the literature. For instance, Fahmy & Blostein [10] proposed a method based on rewriting graphs arising from raw symbol-recognition to produce graphs that conform more closely to musical constraints. Bainbridge & Bell [1] have also proposed making greater use of the semantics of musical notation. Raphael and co-workers [12, 17] have proposed a Bayesian approach based on likelihood, again taking account of musical constraints. Church & Cuthbert [7] proposed correcting errors by using information from similar musical sequences in the same piece of music. Finally, Rossant & Bloch [19] proposed the use of fuzzy logic. Since all of these approaches aim to take an image of a score and output the symbols which are represented in that score, they could be incorporated into the workflow described here, and its output combined with other outputs to further improve recognition.

6. ACKNOWLEDGEMENTS

This work was supported with funding from the Arts and Humanities Research Council (AHRC; AH/L009870/1).

7. REFERENCES

- [1] D. Bainbridge and T. Bell: "A music notation construction engine for optical music recognition," *Software—Practice and Experience*, Vol. 33, pp. 173–200, 2003.
- [2] D. Bainbridge, C. G. Nevill-Manning, I. H. Witten, L. A. Smith, and R. J. McNab: "Towards a digital library of popular music," *Proc. of the 4th ACM conference on Digital Libraries (DL '99)*, pp. 161–169, 1999.
- [3] D. Bainbridge and K. Wijaya: "Bulk Processing of Optically Scanned Music", *Proc. of 7th Int. Conf. on Image Processing And Its Applications*, Vol. 1, pp. 474–478, 1999.
- [4] E. P. Bugge, K. L. Juncher, B. S. Mathiasen, and J. G. Simonsen: "Using sequence alignment and voting to improve optical music recognition from multiple recognisers," *Proc. of the 12th Int. Soc. for Music Information Retrieval Conf.*, pp. 405–410, 2011.
- [5] D. Byrd and M. Schindele: "Prospects for Improving OMR with Multiple Recognisers," *Proc. of the 7th Int. Conf. on Music Information Retrieval (ISMIR)*, pp. 41–46, 2006. Revised and expanded version (2007) retrieved February 20, 2013, <http://www.informatics.indiana.edu/donbyrd/MROMRPap>.
- [6] G. S. Choudhury, T. DiLauro, M. Droettboom, I. Fujinaga, B. Harrington, and K. MacMillan: "Optical Music Recognition System within a Large-Scale Digitization Project," *Proc. of the Int. Symp. on Music Information Retrieval*, 2000.
- [7] M. Church and M. S. Cuthbert: "Improving Rhythmic Transcriptions via Probability Models Applied Post-OMR," *Proc. of the 15th Conf. of the Int. Soc. for Music Information Retrieval*, pp. 643–647, 2014.
- [8] M. S. Cuthbert and C. Ariza: "music21: A Toolkit for Computer-Aided Musicology and Symbolic Music Data," *Proc. of the 11th Int. Soc. for Music Information Retrieval Conf.*, pp. 637–42, 2010.
- [9] C. Dalitz, M. Droettboom, B. Pranzas, and I. Fujinaga: "A Comparative Study of Staff Removal Algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(5), 753–766, 2008.
- [10] H. Fahmy and D. Blostein: "A Graph-Rewriting Paradigm for Discrete Relaxation: Application to Sheet-Music Recognition," *International Journal of Pattern Recognition and Artificial Intelligence*, Vol. 12, No.6, pp. 763–799, 1998.
- [11] M. Good: "MusicXML for notation and analysis," in *The Virtual Score: Representation, Retrieval, Restoration (Computing in Musicology 12)*, W. B. Hewlett, and E. Selfridge-Field, Eds. MIT Press, 2001, pp. 113–124.
- [12] R. Jin and C. Raphael: "Interpreting Rhythm in Optical Musical Recognition," *Proc. of the 12th Int. Soc. for Music Information Retrieval Conf.*, pp. 151–156, 2012.
- [13] I. Knopke and D. Byrd: "Towards Musicdiff: A Foundation for Improved Optical Music Recognition Using Multiple Recognizers," *Proc. Of the 8th Int. Conf. on Music Information Retrieval*, pp. 123–126, 2007.
- [14] H. Miyao and M. Okamoto: "Stave Extraction for Printed Music Scores Using DP Matching," *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 8, No. 2, pp. 208–215, 2004.
- [15] S. B. Needleman and C. D. Wunsch: "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of molecular biology*, Vol. 48, No. 3, pp. 443–453, 1970.
- [16] V. Padilla, A. Marsden, A. McLean, and K. Ng: "Improving OMR for Digital Music Libraries with Multiple Recognisers and Multiple Sources," *Proc. of the 1st Int. Workshop on Digital Libraries for Musicology*, pp. 1–8, 2014.
- [17] C. Raphael and J. Wang: "New Approaches to Optical Music Recognition," *Proc. of the 12th Int. Soc. for Music Information Retrieval Conf.*, pp. 305–310, 2011.
- [18] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. Marcal, C. Guedes, and J. S. Cardoso: "Optical music recognition: state-of-the-art and open issues," *International Journal of Multimedia Information Retrieval*, Vol. 1, No. 3, pp. 173–190, 2012.
- [19] F. Rossant and I. Bloch: "Robust and Adaptive OMR System Including FuzzyModeling, Fusion of Musical Rules, and Possible Error Detection," *EURASIP Journal on Advances in Signal Processing*, Vol. 2007, Article ID 81541, 25 pp.
- [20] N. Saitou and M. Nei: "The neighbor-joining method: a new method for reconstructing phylogenetic trees," *Molecular biology and evolution*, 4(4), 406–425, 1987.
- [21] C. S. Sapp: "Online Database of Scores in the Humdrum File Format," *Proc. Of the 6th Int. Conf. on Music Information Retrieval*, pp. 664–665, 2005.
- [22] S. Suzuki and K. Abe: "Topological Structural Analysis of Digitized Binary Images by Border Following," *Computer Vision, Graphics and Image Processing*, 30(1), 32–46, 1985.
- [23] V. Viro: "Peachnote: Music Score Search and Analysis Platform," *Proc. of the 12th Int. Soc. for Music Information Retrieval Conf.*, pp. 359–362, 2011.
- [24] M. Zvelebil and J. Baum: *Understanding bioinformatics*. Garland Science, Abingdon, 2007.

Relating Natural Language Text to Musical Passages

Richard Sutcliffe

School of CSEE
University of Essex
Colchester, UK
rsutcl@essex.ac.uk

Tim Crawford

Dept of Computing
Goldsmiths, University
of London
t.crawford@gold.ac.uk

Chris Fox

School of CSEE
University of Essex
Colchester, UK
foxcj@essex.ac.uk

Deane L. Root

Department of Music
University of Pittsburgh
Pittsburgh, PA, USA
dlr@pitt.edu

Eduard Hovy

Lang Technologies Inst
Carnegie-Mellon Univ
Pittsburgh, PA, USA
hovy@cmu.edu

Richard Lewis

Department of Computing
Goldsmiths, University of
London
richard.lewis@gold.ac.uk

ABSTRACT

There is a vast body of musicological literature containing detailed analyses of musical works. These texts make frequent references to musical passages in scores by means of natural language phrases. Our long-term aim is to investigate whether these phrases can be linked automatically to the musical passages to which they refer. As a first step, we have organised for two years running a shared evaluation in which participants must develop software to identify passages in a MusicXML score based on a short noun phrase in English. In this paper, we present the rationale for this work, discuss the kind of references to musical passages which can occur in actual scholarly texts, describe the first two years of the evaluation and finally appraise the results to establish what progress we have made.

1. INTRODUCTION

A traditional Information Retrieval (IR) system takes as input a short textual query and a document collection and returns a list of documents which match the query [27]. By combining IR with Natural Language Processing (NLP) the field of Question Answering was born [13], leading to systems which could take a query as input and produce an exact answer [17-20,24]. In the meantime, Music Information Retrieval (MIR) has become a very active area in which various kinds of query are matched against music recordings or electronic forms of score such as MEI [11] (inspired by TEI [25]) or MusicXML [15].

However, music involves text as well as scores; there is a vast body of textual information concerned with Western classical music. First and foremost, Grove's Dictionary of Music and Musicians has developed from a

four-volume printed dictionary published in 1879-1889 into Grove Online which contains around 50,000 signed articles and 30,000 biographies contributed by over 6,000 scholars [6]. In addition, there are countless scholarly books, journal articles and conference papers as well as numerous online sources such as the Wikipedia. All these sources contain detailed analyses of musical works which necessarily make reference to specific passages in scores. Our long-term objective is to investigate whether these references – expressed in a natural language such as English – can be automatically matched to the musical passages to which they refer.

In pursuit of our objective we organised in 2014 [23, 10] and 2015 [to appear] shared evaluations called C@merata (C@ssical Music Extraction of Relevant Aspects by Text Analysis) – <http://csee.essex.ac.uk/camerata/> – in which a number of participants each built a system which could take as input a question in English and a score in MusicXML and identify one or more passages in the score which matched the question. We describe those evaluations and the rationale behind them. We first outline the background to this work and its origins in Question Answering (QA). Second, we present an analysis of text examples, taken from the writings of three important musicologists, which refer to musical passages. Third and Fourth we describe the two C@merata campaigns. Finally we discuss what we have learned and draw some conclusions.

2. BACKGROUND TO OUR EVALUATIONS

Our work is derived from three existing areas of research. First, the considerable body of MIR work concerned with finding passages in music scores based on inputs of various kinds, e.g. [5].

Secondly, the Music Information Retrieval Evaluation Exchange has been organised by J. Stephen Downie since 2005 [4,12]. These landmark evaluations have been concerned with many different tasks over the years and are related to parallel evaluations concerning IR and NLP at TREC [26], CLEF [1] and NTCIR [16]. While MIREX has often been concerned with audio-based systems, it has regularly featured score-based tasks which, in the light of our work, could be combined with natural



© R. Sutcliffe, T. Crawford, C. Fox, D.L. Root, E. Hovy and R. Lewis.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** R. Sutcliffe, T. Crawford, C. Fox, D.L. Root, E. Hovy and R. Lewis. "Relating Natural Language Text to Musical Passages", 16th International Society for Music Information Retrieval Conference, 2015.

language input.

Thirdly, there have been QA tracks at CLEF, starting in 2003 [20]. However, these were not concerned with music until 2011. In that year, the Question Answering for Machine Reading (QA4MRE) task featured difficult multiple-choice questions in four domains, one being Music and Society [18]. Four documents in this domain were used, each taken from transcripts of talks delivered at the TED Conferences. In the 2012 task [19], the music texts used were drawn from Wikipedia, Project Gutenberg and the 1911 Encyclopedia Britannica. Finally, in 2013, the four documents were taken with permission from Grove Online [6]. This gave us the idea of combining text processing with core processing.

3. REFERENCES TO SCORES IN MUSIC TEXTS

In this section, we motivate our work by providing a short description of the references to musical passages in three important text sources. The first is an analysis of the Beethoven Symphonies by Antony Hopkins (Chapter 2: *Symphony No. 1 in C Major Op. 21*) [7] (henceforth ah). The second is the study of Domenico Scarlatti by Ralph Kirkpatrick (Chapter 10: *Scarlatti's Harmony*, Section *Cadential vs. Diatonic Movement of Harmony*) [9] (henceforth rk). The third is the entry for Anton Bruckner by Deryck Cooke (*Section 7. Music*) [2] from the New Grove Dictionary of Music and Musicians [21] (henceforth dc).

We extracted phrases from the above works by hand – 261 in all – and organised them into 14 categories: notes, intervals, scales, melodies, rhythms, tempi, dynamics, keys, harmony, counterpoint, texture & instrumentation, bar numbers, passages & sections and structures & sequences. Furthermore, they are classed as Specific or Vague. Examples of each category can be seen in Table 1, with two Specific and two Vague for each phrase type. The source is indicated in square brackets: [ah26] means ah (i.e. Hopkins) p26; [dc364lh] means dc (i.e. Cooke) p364 in Grove, left hand column.

It is important to note that the categories in Table 1 are for illustration only and are neither exhaustive nor mutually exclusive. The examples are given purely to illustrate the kinds of references to musical passages which one might find in a musicological text. Moreover, the binary categorisation into Specific and Vague is also purely for illustration purposes as specificity lies on a scale. We now draw some conclusions from this table.

The first point to note is that the references vary in specificity; some are clear and unambiguous (C#-D rising semitone, D major, eight-part choir, bars 189-198); others are much more difficult to pin down (alien F#, disturbing syncopations, anguished D minor chromaticism, varied alternation of two long-drawn themes). Secondly, however, all the phrases are meaningful – an expert familiar with the works concerned is likely to be able to identify the points mentioned in the score with a fair accuracy (high Precision even if not necessarily high Recall). This suggests that they are interesting and worthwhile to study.

Thirdly, some categories of phrase lend themselves to

Category	S/V	Examples
Notes	S	[ah26] giant unison G from the entire orchestra [rk220] based on nothing else but A, D, E, and A
	V	[ah12] alien F# in the ascending scale [dc364lh] pedal point
Intervals	S	[ah24] C#-D rising semitone [dc363lh] an ascending diminished fifth
	V	[ah19] fragment of five rising crochets [dc364lh] themes based on falling octaves
Scales	S	[dc363lh] parts entering successively on the degrees of the ascending scale of D major [dc363rh] old church modes ... Phrygian and Lydian
	V	[ah28] the initial scale [ah29] little scales dart to and fro
Melodies	S	[ah13] semiquaver descent in bar 18 [ah19] fragment of five rising crochets
	V	[ah19] Second Subject appearing in the tonic key [dc363rh] the chorale themes in the symphonies
Rhythms	S	[ah15] quaver pattern [ah25] repeated crochets chords
	V	[ah18] disturbing syncopations [dc364lh] hammering ostinatos
Tempi	S	[ah11] slow tempo [dc366lh] slow movements
	V	[ah28] rustic oom-pah bass [dc364rh] intense and long-drawn string cantabile
Dynamics	S	[ah26] violins in bar 126 come in FF [ah29] sudden fortissimo outburst
	V	[ah29] sudden roaring [dc364lh] murmuring tremolando
Keys	S	[ah10] D major [dc366lh] in Bb minor
	V	[rk221] modulatory excursion of the second half [dc363rh] unusual key changes
Harmony	S	[rk221] major dominant [dc364lh] tonic triad of E major
	V	[rk220] departure from three-chord harmony [dc363lh] anguished D minor chromaticism
Counterpoint	S	[ah23] cellos provide a delicate counterturne [dc363lh] parts entering successively on the degrees of the ascending scale of D major
	V	[rk220] dominated by diatonic movement of parts [dc363lh] bold polyphonic imitation of a single point
Texture, Instrumentation	S	[dc363lh] eight-part choir [dc363rh] a piece of unison plainsong
	V	[ah29] decked with garlands of scales from flutes, clarinets and bassoons [dc364rh] a faint background sound, emerging almost imperceptibly out of silence
Bar Numbers	S	[ah15] bars 189-198 [rk220] measure thirteen to measure fifteen
	V	[ah24] sixteen or at most thirty-two bars long [dc365rh] over periods of 16, 32 or even 64 bars
Passages, Sections	S	[dc363rh] whose slow movement and finale [dc364rh] far-ranging first movement
	V	[rk220] series of small sequential passages [dc362rh] a passage from the Gloria
Structures, Sequences	S	[ah18] First Subject [rk221] Phrygian cadence
	V	[dc365rh] exposition (nearly always built on three subject groups rather than two) [dc366rh] varied alternation of two long-drawn themes

Table 1. Fourteen types of referring expressions, categorised into Specific (S) and Vague (V).

rather simple and clear expression. Examples include Notes (G), Intervals (ascending diminished fifth), Scales (D major), Rhythms (repeated crochets chords), Dynamics (FF), Keys (Bb minor) and bar numbers (measure thirteen). If we set ourselves the task of

searching for such passages in a score, we are likely to be quite successful.

Fourthly, some categories of phrase tend conversely to be complex and often imprecise as well. Examples include Texture & Instrumentation (a faint background sound, emerging almost imperceptibly out of silence), Passages & Sections (a passage from the Gloria) and Structures & Sequences (exposition (nearly always built on three subject groups rather than two)). Western classical music excels in structure and in harmony, so treatment of these topics tends to be particularly interesting and important. The richness and ambiguity of language are its strengths in this context as a great deal can be suggested in relatively few words. Moreover, to the expert, the references remain quite clear, though a considerable amount of knowledge and background information is being brought to bear.

Fifthly, it is interesting to observe that many of the examples in Table 1 are noun phrases; this construct can express very complicated and detailed concepts in a musicological text.

Sixthly and finally, phrases in natural language can never be replaced by expressions in a pattern language (such as regular expressions applied over text strings). Such expressions are by their nature unambiguous and in practical contexts they are usually concise. Therefore, the study of natural language in musicology is not made unnecessary by the existence of such languages. On the other hand, such expression languages are extremely useful and worthwhile [28]; one possible application of them here is to map a natural language phrase onto a pattern (possibly extremely complex) in such an expression language in order to initiate a search.

In the next section we will describe our evaluations.

4. THE 2014 C@MERATA TASK

4.1 Input Provided

In a QA evaluation such as ResPubliQA [17], the input is normally a short question such as ‘Who is President of the United States’ and the output is an exact answer such as ‘Barack H. Obama’. As we have discussed earlier, many of the real examples in Table 1 are in fact noun phrases. So it seemed reasonable to use a noun phrase as the input for an initial evaluation, rather than a complete question. The top of Table 2 shows the question types which were adopted. For all the types mentioned below, there are several examples in the right hand column.

As we observed above, entries in the Notes category of Table 1 are some of the simplest and clearest. As this was a new task, it was decided to include three simple query types in the evaluation which correspond broadly to Note: simple_pitch, simple_length and pitch_and_length.

Perf_spec queries combine a note with some performance indication. Stave_spec queries restrict the answer to a particular stave in the score which may be specified in various ways, including the instrument concerned, the hand being used (for keyboard music) or the clef on which the music appears. Similarly, word_spec queries link a note to the word which is sung on it in one of the parts.

Question Types for 2014 Task		
Type	No	Examples
simple_pitch	30	G5, E, A natural, C flat, F#4, F2 sharp
simple_length	30	dotted quarter note, quarter note rest, semiquaver rest, whole note, semibreve
pitch_and_length	30	D# crotchet, half note C, quarter note B5, semiquaver G#, half note Db, quaver F#
perf_spec	10	D sharp trill, fermata A natural, staccato B flat, marcato D flat, F trill, down bow E
stave_spec	20	D4 in the right hand, half note D in the viola, treble clef A sharp, F3 sharp in the “alt”, quarter note F in the Alto
word_spec	5	word “Se” on an A flat, minim on the word “Der”, minim B on the word “im”, G on the word “praise”
followed_by	30	crotchet followed by semibreve, D followed by G, quarter note G followed by eighth note G, dotted quaver E followed by semiquaver F sharp, crotchet rest followed by crotchet, dotted quarter note followed by A4
melodic_interval	19	melodic octave, rising major sixth, melodic descending fifth, falling major third, melodic rising minor third, octave leap, falling tone, melodic fourth
harmonic_interval	11	harmonic major sixth, harmonic second, nineteenth, seventh, harmonic fifth, harmonic octave, major seventeenth
cadence_spec	5	perfect cadence
triad_spec	5	tonic triad, lb triad, triad in first inversion, la triad
texture_spec	5	polyphony, melody with accompaniment, monophony, homophony
All	200	
Question Types for 2015 Task		
Type	No	Examples
l_melod	40	D4 minim, eighth note in measure 9
l_melod qualified by perf, instr, clef, time, key	40	trill on a quaver A; G# in the Cello part in measures 29-39; sixteenth note # in the left hand; half note E3 in 2/2; sixteenth note G in G minor in measures 1-5
n_melod	20	F# E G F# A; Do Mi Do Sol Do Mi Sol Do in bars 1-20; twenty semiquavers; five note melody in bars 1-10
n_melod qualified by perf, instr, clef, time, key	20	two staccato quarter notes in the Violin I; crotchet, crotchet rest, crotchet rest, crotchet, crotchet rest, major, crotchet, crotchet, crotchet, crotchet in the Timpani; melodic octave leap in the bass clef in measures 70-80; G4 B4 E5 in 3/4; rising G minor arpeggio
l_harm possibly qualified by perf, instr, clef, time, key	20	eighth note chord Bb, C, E; chord of D minor in measures 109-110; harmonic minor sixth in the Violas; dotted minim chord in the left hand
texture	6	monophonic passage; homophony in measures 1-14; polyphony in measures 10-14; Alberti bass in measures 0-4
follow possibly qualified on either or both sides by perf, instr, clef, time, key	40	quavers F4 E4 in the oboe followed by quavers E2 G#2 in the bass clef; quarter note minor third followed by eighth note unison; C followed by mordent Bb; chord C4 G4 C5 E5 then a quaver; three eighth notes in the Violin I followed by twelve sixteenth notes in the Violin II in measures 87-92
synch possibly qualified in either or both parts by perf, instr, clef, time, key	14	four eighth notes against a half note; crotchet D3 on the word “je” against a minim D2; four staccato quavers in the Violoncello against a minim chord Ab3 C4 F4 in the Harpsichord
All	200	

Table 2. Summary of question types in tasks.

- Q: G flat
- A: [4/4, 2, 67:5-67:5], [4/4, 2, 71:2-71:2]
- Q: semibreve
- A: [4/4, 1, 76:1-76:4]
- Q: minim F
- A: [4/4, 1, 67:1-67:2]
- Q: minim C in the bass
- A: [4/4, 1, 72:1-72:2], [4/4, 1, 72:3-72:4]
- Q: crotchet followed by semibreve
- A: [4/4, 1, 75:4-76:4]
- Q: melodic octave
- A: [4/4, 2, 69:5-69:8], [4/4, 2, 72:1-72:8], [4/4, 2, 73:5-73:8], [4/4, 2, 74:5-74:8], [4/4, 2, 75:5-75:8]

Figure 1. Extract from Scarlatti K466 with questions and answers from the 2014 task.

So far, all the query types are simple notes in isolation. Queries of type followed_by specify two adjacent notes.

As Table 1 showed, intervals are discussed in real texts, so we wished to include some queries of this type. We divided them into two kinds, melodic and harmonic. melodic_interval specifies two adjacent notes on the same staff which are a specified distance apart. Conversely, a harmonic_interval specifies two simultaneous notes. Unlike melodic intervals, harmonic intervals were permitted to occur across staves because they are integral to the concept of harmony which is often created by instruments or voices in different parts. Intervals are considered harmonic by default, thus ‘fifth’ is assumed to be a harmonic fifth.

The last three question types were more experimental, though still being relatively straightforward and unambiguous in musical terms. cadence_spec requires a cadence to be identified. A triad_spec specifies triads in various forms of notation. Finally, texture_spec states the required texture to be found. Referring back to Table 1, cadences touch upon Structures & Sequences and Triads are a fundamental element of Harmony.

There were 200 queries in a fixed distribution as shown in the middle column of Table 2. The four simplest query types (simple_pitch, simple_length, pitch_and_length, followed_by) were the most numerous in the test set with 30 each. After this came stave_spec and melodic interval with twenty each followed by perf_spec and harmonic_interval with ten each. (One melodic interval was changed for a harmonic_interval at a late stage, so in fact there were nineteen of the former

- Q: dotted minim F#4
- A: [3/4, 1, 65:1-65:3]
- Q: F4 crotchet in the oboe
- A: [3/4, 2, 64:3-64:4]
- Q: minim A2 in 3/4 time
- A: [3/4, 1, 62:2-62:3], [3/4, 1, 64:2-64:3]
- Q: chord D2 E5 G5 in bars 54-58
- A: [3/4, 2, 57:1-57:1]
- Q: quavers F3 A3 followed by crotchet A4 in the violin
- A: [3/4, 1, 57:2-57:3]
- Q: four quavers in the violin against a minim in the bass clef
- A: [3/4, 1, 62:2-62:3], [3/4, 1, 64:2-64:3]

Figure 2. Extract from Bach BWV1047 Andante with questions and answers from the 2015 task.

and eleven of the latter.) Finally, there were five each of word_spec, cadence_spec, triad_spec and texture_spec. Thus some more experimental types of query were represented in the task but played a relatively minor role.

In summary, most of the question types used in 2014 were straightforward and were derived from Notes, Intervals and (partly) Harmony, Texture & Instrumentation and Structures & Sequences. Other phrase types of Table 1 were not catered for.

4.2 Output Required

As we have seen, an input query was simply a short noun phrase. To make the evaluation as simple as possible, an answer was defined to be a subsection of a score, starting and ending at a particular place. The answer was not required to specify which stave (or staves) contained the answer.

Initially, we planned to measure beats in a bar in terms of the shortest note (hemidemisiquaver, one sixteenth of a crotchet). However, this does not allow for triplets (where, say, a crotchet is divided into three) or any other sort of n-tuplet. So instead, we adopted the concept of divisions from MusicXML. The divisions value is the number of beats into which the crotchet is divided. A suitable value depends on what we wish to demarcate as an answer. So for simplicity, we specified for each query the divisions value to be used for the answers.

Based on these ideas we developed the concept of a passage which would contain, for both start and end, a time signature, a divisions value, and a bar and beat.

The start bar and beat is where the passage is defined to commence. More precisely, the passage begins in the denoted bar immediately *before* the start beat, measured from the beginning of the bar in the unit of time denoted by the stated divisions value. Similarly, the passage is defined to end immediately *after* the end beat. We adopted this before-the-start and after-the-end after careful thought and discussion. The advantage of it is that it is intuitive: As can be seen in Figure 1, above, the first two crotchets in bar 67 are denoted 67:1-67:2 which can be understood at a glance.

We developed three equivalent ways of stating a passage: Ascii Long Form, Ascii Short Form and XML form. The Ascii forms are convenient for discussions in papers etc. while the XML form is useful as the input to, and output from programs.

Here is an example in short form: [4/4,1,1:1-2:4]. The time signature is 4/4 and divisions value is 1. The passage starts in bar 1 before the first crotchet (i.e. 1:1) and ends in bar two after the fourth crotchet (i.e. 2:4). We take bar numbers from the MusicXML score.

We use the XML format for specifying the test queries for participants as well as for the queries plus correct answers (often called the Gold Standard in QA).

In summary, our passage specifies two vertical lines drawn through the score and does not distinguish between the different staves. We thus assume that any answer can be exactly demarcated in this way. We will return to this point in the conclusions.

4.3 Evaluation

Precision, Recall and F-Measure are commonly used in IR and NLP [27]. We wished to determine all the correct answer passages by hand to produce a Gold Standard and then to compare the results returned by a system to that.

It is useful to have both strict and lenient measures in an evaluation. At the fourth TREC QA track onwards (starting in 2002) there were four judgements of each answer, Right, ineXact, Unsupported and Wrong [29]. In the TREC context a correct answer could be 'Bill Clinton' while an ineXact one could be 'Clinton' or perhaps 'Bill Clinto'. Unsupported answers were Right but not shown to be so from a document in the collection.

We decided that a passage returned which began at the right bar and beat within the bar and also ended at the right bar and beat within the bar was correct. On the other hand, an answer which started and ended at the right bar (but not necessarily the right beat in the bar) was still very useful and could be considered the equivalent of TREC's ineXact. If an expert is looking for a particular cadence, for example, and is told the bar numbers, they can usually see it at a glance. However, searching through hundreds of bars looking for the cadence is time consuming. The concept of Unsupported is not applicable to our task. The measures were thus defined as follows:

Beat Precision (BP) is the number of beat-correct passages returned by a system divided by the number of passages (correct or incorrect) returned.

Beat Recall (BR) is the number of beat-correct passages returned by a system divided by the total number of answer passages known to exist.

2014 Results	BP	BR	BF	MP	MR	MF
Maximum	0.713	0.904	0.797	0.764	0.967	0.854
Minimum	0.113	0.150	0.185	0.155	0.154	0.226
Average	0.420	0.654	0.483	0.460	0.734	0.534
2015 Results	BP	BR	BF	MP	MR	MF
Maximum	0.817	0.739	0.620	0.817	0.809	0.656
Minimum	0.061	0.175	0.108	0.073	0.175	0.129
Average	0.351	0.564	0.348	0.370	0.619	0.375

Table 3. Results of the 2014 & 2015 tasks.

As is usual, **Beat F-Score (BF)** is the harmonic mean of BP and BR.

Measure Precision (MP) is the number of bar-correct passages returned by a system divided by the number of passages (correct or incorrect) returned.

Measure Recall (MR) is the number of bar-correct passages returned by a system divided by the total number of answer passages known to exist.

Finally, **Measure F-Score (MF)** is the harmonic mean of MP and MR.

4.4 Scores

After consideration of several notations including kern [8], MusicXML was chosen because it is widely used and is supported by music21 [3] and musescore [14].

Twenty MusicXML scores were used and ten questions were set on each, forming the question type distribution of Table 2. We incorporated both European (crotchet, bar etc) and American (quarter note, measure etc) terms into the task by setting American queries for ten of the twenty scores and English queries for the rest.

Scores for 2014 were chosen from the Renaissance and Baroque in order to avoid more heavily-scored works from the Classical period onwards. The composers chosen were Bach, Carissimi, Charpentier, Corelli, F. Cutting, Dowland, Lully, Monteverdi, Purcell, A. Scarlatti, D. Scarlatti, Tallis, Telemann, Vivaldi and S. L. Weiss. Scores were chosen on a predefined distribution: six on two staves, six on three staves, four on one staff and two each on four staves and five staves. There were works for solo cello, harpsichord and lute; one, three, four and five voices; soprano or cello and harpsichord; two violins and cello; two violins, viola and two cellos.

The scores were obtained from two sources. Most came from musescore.com. Two Bach chorales were used and both came from www.jsbchorales.net. We required scores to have a license 'to share' rather than just 'for personal use'. Moreover, we required scores to be well presented, transcribed in a scholarly manner and provided in valid MusicXML Version 2 or lower.

4.5 Questions

Each score was sent to one of the organisers who was asked to set questions according to the target distribution of Table 2. It was specified for each score whether the questions were to be in American or English. For each question, answers were to be provided in the Ascii short form for specifying passages. The organiser in question

was asked to find all answers for all the questions. The question data was returned in an Ascii format which incorporates the score filename, the questions, the answers in Short Ascii form and also any comments concerning the questions or answers.

On receipt of the files, the questions and answers were checked by a second expert who noted any changes or observations using comments in the Ascii file. The second expert also carried out an independent search for answer passages within the scores. When all changes were checked and validated, the complete set of twenty Ascii files was transformed automatically into XML format in order to form the Gold Standard for the task.

4.6 Participants, Runs and Results

The task was announced in January 2014. Five participants registered; two were from Ireland and the other three came from Australia, England and India. Participants had one week to complete their runs starting from 16th June 2014.

Each participant was allowed to submit up to three runs. The overall results are shown in Table 3. The best BF (strict) score was 0.797 which was remarkably good.

Averages for BF and MF are 0.483 and 0.534 so systems scored better under lenient measures than under strict measures but the difference is not large – only 11%. Concerning the top run, the difference between MF=0.854 and BF=0.797 is only 7%. So if a system finds the correct bar, it tends to find the exact beat in the bar as well. Generally, the average figures suggest that participants had all made a very good attempt at building a system for this very complicated task.

4.7 Approaches to the Task

Concerning software, most participants opted to use Python and to adapt a baseline system using music21 [3] which we wrote and distributed [22]. Others used their own tools in Lisp or C.

Only basic NLP was used. Typically the query was scanned looking for terms (e.g. down bow) and converting them to concepts (down_bow). Some systems adopted a QA approach and assigned the query to a pre-define set of types, each with its method of solution. Others converted the concepts to a structured representation by parsing the concepts. The final stage was a search of the score. Some varied the representation of the score according to the query type (e.g. using music21 chordify for cadence questions). As all answers to a given query were defined to lie in exactly one of the scores, no one opted to use any inverted indexing of the music data.

5. THE 2015 C@MERATA TASK

5.1 Changes from 2014

This year's campaign has just concluded. The use of MusicXML scores, the XML formats for questions and answers, the passage concept and the evaluation measures remained the same in 2015. However, there was a wider range of score types from the Renaissance to the early Romantic periods, scores were more complicated – up to

nineteen staves – and questions were differently organised and generally more difficult (see Table 2). For example, an n_melod question can specify quite complicated melodies while the synch type can link two simultaneous features.

5.2 Participants, Runs and Results

The same five participated as in 2014. The maximum BF was 0.620 and the average BF was 0.348 (Table 3), both lower than last year. However, the task was considerably harder and the participants did very well.

6. DISCUSSION AND CONCLUSIONS

First, in both years, participants were able to build a working system and submit valid runs.

Second, all systems could make a good attempt at answering at least one of the question types.

Third, the best systems (see Table 3) achieved very good results and several others were not far behind.

Fourth, the technical basis of the task was shown to be sound and all the steps of the campaigns were fulfilled.

Fifth, the development of strict measures (BP, BR, BF) and lenient measures (MP, MR, MF) specifically for this task worked well.

Sixth, the ability to evaluate runs automatically showed the practicality and scalability of the evaluation.

There were also some shortcomings; first, our passage concept does not distinguish between staves. Suppose a minim F starts in the first beat of bar 1 in the treble clef and in the second beat of bar 1 in the bass clef (of a keyboard work). The two answer passages thus overlap which is anomalous. On the other hand, consider a texture such as homophony where some instruments have rests for some or all of the passage – are those instruments part of the passage or not?

Second, not all passages of interest in a score can be demarcated exactly. For example, a polyphonic passage may commence in a madrigal when a homophonic section is still drawing to a close. If we say 'most' parts must be participating in polyphony is that the start of it, or must 'all' participate? Also, what about the start and end of a triad? Sometimes the bass note is only established after the other notes.

Third, some 'passages' may turn out to have no length. Consider the perfect cadence. The V chord can be set up in many different ways such that it can be hard to say where exactly that chord starts. Then, the onset of the I chord can be equally ambiguous: there may be a trill on the V; or the I in the treble may be set up either before or after the bass moves to I. For the future, we would consider defining a perfect cadence as a point in a score, not a passage; the instant the bass moves from V to I.

Finally, consider again Table 1 (real phrases) against Table 2 (actual phrases used in 2014 and 2015). There is a considerable difference in complexity and subtlety. Many of our queries were simple notes which present few problems for either NLP or MIR. Future campaigns can include more complex query types which delve further into the subtleties of musical language while still being practical for use in MIR.

7. REFERENCES

- [1] CLEF (2014). <http://www.clef-initiative.eu/>.
- [2] Cooke, D. (1995). Bruckner, (Joseph) Anton. In S. Sadie (ed), *New Grove Dictionary of Music and Musicians*, Volume 3, Section 7. Music (p362-366). London, UK: Macmillan.
- [3] Cuthbert, M. S., & Ariza C. (2010). music21: a toolkit for computer-aided musicology and symbolic music data. *Proc. International Symposium on Music Information Retrieval (Utrecht, The Netherlands, August 09 - 13, 2010)*, p637-642.
- [4] Downie, J. S. (2008). The Music Information Retrieval Evaluation Exchange (2005-2007): A window into music information retrieval research. *Acoustical Science and Technology* 29 (4): 247-255. Available at: <http://dx.doi.org/10.1250/ast.29.247>
- [5] Ganseman, J., Scheunders, P., & D'haes, W. (2008). Using XQuery on MusicXML databases for musicological analysis. *Proc. International Symposium on Music Information Retrieval*, p433-438.
- [6] Grove Music Online (2015). <http://www.oxfordmusiconline.com/public/>
- [7] Hopkins, A. (1982). *The Nine Symphonies of Beethoven*. London: Pan Books.
- [8] Huron, D. (1997). Humdrum and Kern: Selective Feature Encoding. In 'Beyond MIDI', ed. E. Selfridge-Field (p375-401). Cambridge, MA: MIT Press.
- [9] Kirkpatrick, R. (1953). *Domenico Scarlatti*. Princeton, NJ: Princeton University Press.
- [10] Larson, M., Riegler, M. A., Miro, X. A., Korshunov, P., Petkos, G., Soleymani, M., Choi, J., Schedl, M., Ionescu, B., Eskevich, M., Jones, G., & Sutcliffe, R. F. E. (2014). *Proc. MediaEval 2014 Workshop*, Barcelona, Spain, October 16-17 2014. <http://ceur-ws.org/Vol-1263/>.
- [11] MEI (2014). *Music Encoding Initiative*. <http://music-encoding.org/home>.
- [12] Mirex (2014). http://www.music-ir.org/mirex/wiki/MIREX_HOME
- [13] Mollá, D., & Vicedo, J. L. (2007). Question answering in restricted domains: An overview. *Comput. Linguist.*, 33(1):41-61.
- [14] Musescore (2014). *Music Composition and Notation Software*. <http://musescore.org/>.
- [15] MusicXML (2014). <http://www.musicxml.com/>.
- [16] NTCIR (2014). <http://research.nii.ac.jp/ntcir/index-en.html>.
- [17] Peñas, A., Forner, P., Sutcliffe, R., Rodrigo, A., Forascu, C., Alegria, I., Giampiccolo, D., Moreau, N., & Osenova, P. (2009). Overview of ResPubliQA 2009: Question Answering Evaluation over European Legislation Notebook of the Cross Language Evaluation Forum, CLEF 2009, Corfu, Greece, 30 September - 2 October.
- [18] Peñas, A., Hovy, E., Forner, P., Rodrigo, A., Sutcliffe, R., Forascu, C., Sporleder, C. (2011). Overview of QA4MRE at CLEF 2011: Question Answering for Machine Reading Evaluation. *Proc. QA4MRE-2011*. Held as part of CLEF 2011.
- [19] Peñas, A., Hovy, E., Forner, P., Rodrigo, A., Sutcliffe, R., Sporleder, C., Forascu, C., Benajiba, Y., Osenova, P. (2012). Overview of QA4MRE at CLEF 2012: Question Answering for Machine Reading Evaluation. *Proc. QA4MRE-2012*. Held as part of CLEF 2012.
- [20] Peñas, A., Magnini, B., Forner, P., Sutcliffe, R., Rodrigo, A., & Giampiccolo, D. (2012). Question Answering at the Cross-Language Evaluation Forum 2003-2010. *Language Resources and Evaluation Journal*, 46(2), 177-217.
- [21] Sadie, S. (eds) (1995). *The New Grove Dictionary of Music and Musicians*. London, UK: Macmillan.
- [22] Sutcliffe, R. F. E. (2014). A Description of the C@merata Baseline System in Python 2.7 for Answering Natural Language Queries on MusicXML Scores. University of Essex Technical Report, 21st May, 2014.
- [23] Sutcliffe, R. F. E., Crawford, T., Fox, C., Root, D. L., & Hovy, E. (2014). The C@merata Task at MediaEval 2014: Natural language queries on classical music scores. *Proc. MediaEval 2014 Workshop*, Barcelona, Spain, October 16-17 2014. <http://ceur-ws.org/Vol-1263/>.
- [24] Sutcliffe, R., Peñas, A., Hovy, E., Forner, P., Rodrigo, A., Forascu, C., Benajiba, Y., & Osenova, P. (2013). Overview of QA4MRE Main Task at CLEF 2013. *Proc. QA4MRE-2013*.
- [25] TEI (2014). *Text Encoding Initiative*. <http://www.tei-c.org/index.xml>.
- [26] TREC (2014). <http://trec.nist.gov/>.
- [27] van Rijsbergen, K. J. (1979). *Information Retrieval*. London, UK: Butterworth. <http://www.dcs.gla.ac.uk/Keith/Preface.html>
- [28] Viglianti, R. (2015). *Enhancing Music Notation Addressability*. <http://mith.umd.edu/research/project/enhancing-music-notation-addressability/>.
- [29] Voorhees, E. M. (2002). Overview of the TREC 2002 Question Answering Track. <http://trec.nist.gov/pubs/trec11/papers/QA11.pdf>

MUSIC BOUNDARY DETECTION USING NEURAL NETWORKS ON COMBINED FEATURES AND TWO-LEVEL ANNOTATIONS

Thomas Grill and Jan Schlüter

Austrian Research Institute for Artificial Intelligence (OFAI), Vienna, Austria

thomas.grill@ofai.at, jan.schluter@ofai.at

ABSTRACT

The determination of structural boundaries is a key task for understanding the structure of a musical piece, but it is also highly ambiguous. Recently, Convolutional Neural Networks (CNN) trained on spectrogram features and human annotations have been successfully used to tackle the problem, but still fall clearly behind human performance. We expand on the CNN approach by combining spectrograms with self-similarity lag matrices as audio features, thereby capturing more facets of the underlying structural information. Furthermore, in order to consider the hierarchical nature of structural organization, we explore different strategies to learn from the two-level annotations of main and secondary boundaries available in the SALAMI structural annotation dataset. We show that both measures improve boundary recognition performance, resulting in a significant improvement over the previous state of the art. As a side-effect, our algorithm can predict boundaries on two different structural levels, equivalent to the training data.

1. INTRODUCTION

The decomposition of a piece of music into parts known as movements, phrases, chorus and verse, etc., also commonly referred to as *musical form*, is an important task and a major challenge in music analysis. However, the identification and exact placement of transition points, or *boundaries* between such structural elements is often indistinct, even for trained human annotators. Figure 1 represents an excerpt of the piece “The Wet Spot” by “Southern Culture On The Skids” (index 1358 in the SALAMI collection, see Section 4.1). Two different sets of human-annotated boundaries (*ground truth*) are depicted by vertical marks at the top and bottom of the plots. They clearly illustrate the ambiguity of annotating boundaries at a certain level of detail. The annotators agreed well on the positions of the boundaries, but for some of these they disagreed whether they should be considered strong (or ‘coarse’, delimiting

‘large scale’, resp., ‘functional’ sections)¹ or weak (‘fine’, delimiting ‘small scale’ sections). This poses a problem as the common methodology used for the evaluation of structural annotation ignores the hierarchical nature and considers only one level of detail, usually the coarse boundaries.

The currently by far best-performing methods for boundary detection use Convolutional Neural Networks (CNNs), trained on large corpora of human-annotated structural annotations. The algorithms are based on mel-scaled log-magnitude spectrograms (MLSs), taking into account a relatively short context of a few seconds, depending on the desired precision. As shown in Figure 1a, the CNN based solely on an MLS or a variation such as MLS-HPSS (Harmonic-Perceptive Source Separation, see [1]), has difficulties of identifying certain boundaries, indicated by low probabilities in the prediction curve (Figure 1b). We have investigated in [3] that *self-similarity lag matrices* (SSLMs, see Figures 1c and 1d) can be used as additional alternative structural information to significantly improve boundary detection.

In this contribution, we expand on our approach by combining more input features, and put particular focus on the integration of multiple and two-level annotation ground-truth, as available in the SALAMI dataset. The structure of the paper is as follows: After giving an overview over related work in Section 2, we propose our method in Section 3. In Section 4, we describe the experimental setup and our evaluation strategy. Section 5 presents our main results. We wrap up in Section 6 with a discussion and outlook.

2. RELATED WORK

Following the overview paper by Paulus et al. [12], three fundamental approaches to music structure analysis can be distinguished: Novelty-based, detecting transitions between contrasting parts, homogeneity-based, identifying sections that are consistent with respect to their musical properties, and repetition-based, building on the determination of recurring patterns. Novelty is typically computed from self-similarity matrices (SSMs) or self-distance matrices (SDMs) by sliding a checkerboard kernel along the diagonal [2], building on audio descriptors like MFCCs, pitch class profiles, or rhythmic features [10]. Turnbull



© Thomas Grill and Jan Schlüter.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Thomas Grill and Jan Schlüter, “Music boundary detection using neural networks on combined features and two-level annotations”, 16th International Society for Music Information Retrieval Conference, 2015.

¹ See [16] and SALAMI Annotator’s Guide, <http://www.music.mcgill.ca/~jordan/salami/SALAMI-Annotator-Guide.pdf>, accessed 2015-05-04

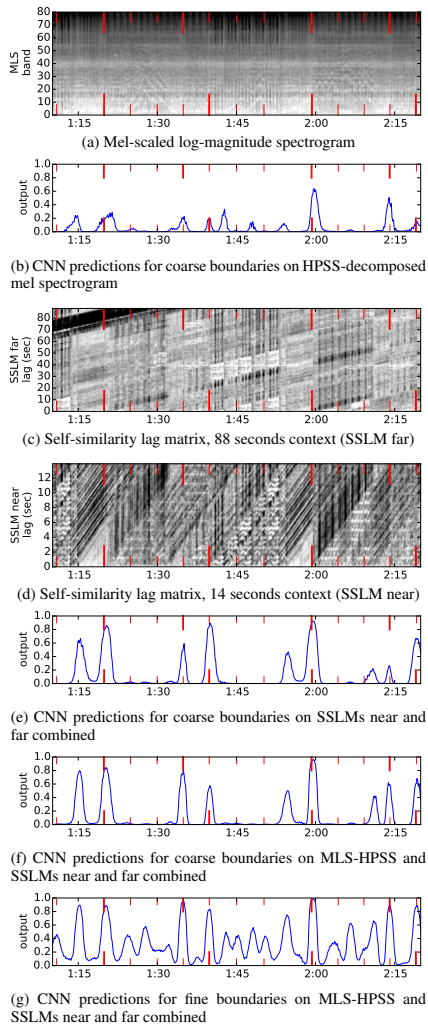


Figure 1: Boundary recognition using CNNs on different underlying audio features, illustrated on the piece “The Wet Spot” by “Southern Culture On The Skids”. Two sets of human annotation ground-truth are shown in red on top and bottom of each plot. Coarse boundaries are thick, fine boundaries are thin. Visit <http://www.ofai.at/research/impml/projects/audiostreams/ismir2015> for a version with audio.

et al. [17] compute difference features on more complex audio feature sets and use trained Boosted Decision Stumps for boundary detection. In order to capitalize on repeated patterns, SSMs or SDMs are used with various heuristic rules and optimization schemes for structure formation [4, 9, 11]. McFee and Ellis employ spectral clustering [6], or add a supervised learning scheme using ordinal linear discriminant analysis and constrained clustering [5]. When using end-to-end neural network techniques such as Ullrich et al.’s CNNs [18], the separation between the fundamental approaches becomes blurred as the CNN infers the relationships between audio features and ground truth from the provided training data. In a similarly integral fashion, Serrà et al. [15] propose an unsupervised method explicitly combining all three domains.

3. PROPOSED METHOD

Our approach is derived from the work by Ullrich et al. [18]. In the following, we will mainly describe our extensions to this method.

3.1 Feature extraction

For each audio file under analysis, we first compute a STFT magnitude spectrogram with a window size of 46 ms (2048 samples at 44.1 kHz sample rate) and 50% overlap, and apply a mel-scaled filterbank of $n = 80$ triangular filters from 80 Hz to 16 kHz and scale magnitudes logarithmically.

From this MLS we compute a HPSS decomposition with a kernel size of 21×21 bins. Preliminary experiments showed that the actual size is a rather insensitive parameter. We either use MLS only or MLS-HPSS (two parallel channels) as one part of the network input.

Our method of generating the SSLMs, which represent similarities of the MLS at one point in time in relation to points in the past, up to a certain *lag time*, is derived from work by Serrà et al. [15] and described in detail in [3]. We use the MLS time series $\mathbf{x}_{i=1..N}$ from above, down-sample it by max-pooling of a factor $p = 2$, and apply a DCT-II transformation on each frame with the static component omitted. Several of these frames are concatenated within a local time context of L bins, equivalent to 0.1 seconds, resulting in the time series $\tilde{\mathbf{x}}_i$. A cosine distance function $\delta_{\cos}(\mathbf{x}, \mathbf{y}) = 1 - \frac{\langle \mathbf{x}, \mathbf{y} \rangle}{\|\mathbf{x}\| \cdot \|\mathbf{y}\|}$ is used to build the $\lfloor \frac{N}{p} \rfloor \times \lfloor \frac{L}{p} \rfloor$ recurrence matrix

$$D_{i,l} = \delta_{\cos}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_{i-l}), \quad l = 1 \dots \lfloor \frac{L}{p} \rfloor. \quad (1)$$

To reveal relationships between distances across this matrix, adaptive thresholding is performed with a smooth sigmoid transfer function $\sigma(x) = 1/(1 + e^{-x})$, yielding

$$R_{i,l} = \sigma\left(1 - \frac{D_{i,l}}{\varepsilon_{i,l}}\right). \quad (2)$$

The adaptive threshold, or, in this context, equalization factor $\varepsilon_{i,l}$ is set to a quantile Q_κ with $\kappa = 0.1$ of the distances $\delta_{\cos}(\tilde{\mathbf{x}}_i, \tilde{\mathbf{x}}_{i-j})$ and $\delta_{\cos}(\tilde{\mathbf{x}}_{i-l}, \tilde{\mathbf{x}}_{i-l-j})$ for $j = 1 \dots \lfloor \frac{L}{p} \rfloor$,

or

$$\varepsilon_{i,t} = Q_{\kappa} \left(D_{i,1}, \dots, D_{i, \lfloor \frac{N}{p} \rfloor}, D_{i-l,1}, \dots, D_{i-l, \lfloor \frac{N}{p} \rfloor} \right). \quad (3)$$

All indices $i < 1$ are wrapped around to $i' = i + \lfloor \frac{N}{p} \rfloor$, resulting in a time-circular SSLM.

3.2 Feature preprocessing

Like [18], for the MLS features, we pad the spectrogram with pink noise of -70 dB FS as needed to process the beginning and end of a piece. For the MLS-HPSS variant, the harmonic and percussive components are separated at this point. After subsampling the MLS by taking the maximum over 6 adjacent time frames without overlap (max-pooling), we normalize to zero mean and unit variance for each frequency band. For the SSLM features, we use circular padding and pooling factors examined in [3]: A factor of 3 for a time context of 14 seconds (feature ‘SSLM-near’), and a factor of 19 for a context of 88 seconds (feature ‘SSLM-far’). We then also normalize each lag band to zero mean and unit variance.

3.3 Convolutional neural network

CNNs are feed-forward networks that include *convolutional layers* computing a convolution of their input with small learned filter kernels of a given size. This allows processing large inputs with few trainable parameters, and retains the input’s spatial layout. When used for binary classification, the network usually ends in one or more dense layers integrating information over the full input at once, discarding the spatial layout. Our architecture for this work is based on the one used by Ullrich et al. [18] on MLS features for their MIREX submission [14]. It has a convolutional layer of $32 \times 8 \times 6$ kernels (8 time frames and 6 frequency bands), a max-pooling layer of 3×6 , another convolution of $64 \times 6 \times 3$ kernels, a dense layer of 128 units and a dense output layer of 1 unit.

We employ a variant of this architecture to support multiple input features instead of one. A comparison of different architectural variations has been shown in [3], where a late ‘time-synchronous fusion’ of the input features, performed in the last convolutional layer, yielded the best results: since the input features cover the same temporal context at the same resolution, their feature maps can be synchronously convolved over time. Figure 2 shows the underlying CNN architecture used for all experiments in our study. The inputs (bottom) are varied, e.g., MLS only is used instead of MLS-HPSS, or one of the input legs is left out. For the outputs (top), either only the coarse unit is used, or both coarse and fine.

Training is done by mini-batch gradient descent, using the same hyper-parameters and tweaks as Ullrich et al. [18]. Likewise, we follow the peak-picking strategy described therein to retrieve likely boundary locations from the network output.

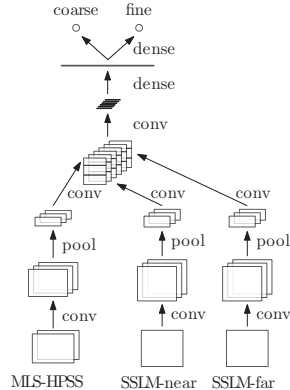


Figure 2: The CNN architecture in use for all the models. The full model is shown here, inputs or outputs were varied for the different experiments.

4. EXPERIMENTS

4.1 Data set

We base our experiments on the data set described by Ullrich et al. [18] which is a subset of the Structural Analysis of Large Amounts of Music Information (SALAMI) [16] version 1.2 database. A part of this SALAMI 1.2 data set was also used in the ‘Audio Structural Segmentation’ task of the annual MIREX evaluation campaign in the years 2012 through 2014.² Lately, the data set has been updated to version 2.0³ with a large number of issues fixed. The entire data set contains over 1600 musical recordings of different genres and origins. In SALAMI version 2.0, a total of 1164 recordings (with 763 double-annotated) are publicly available. Identically to [18], we used 633 musical pieces for training, 100 for validation and 487 pieces as a test set for final evaluation of our models against the published results of the various MIREX submissions.

4.2 Evaluation

For the MIREX campaign’s boundary retrieval task, three different evaluation measures are used: *Hit rate* for time tolerances ± 0.5 and ± 3 seconds, and *Median deviation*. The latter computes the median time distance between each annotated boundary and its closest predicted boundary, and vice versa. The former checks which predicted boundaries fall close enough to an unmatched annotated boundary (true positives), records remaining unmatched predictions and annotations as false positives and negatives, respectively, and computes the precision, recall and F_1 scores. The Hit rate F_1 score is the measure most frequently used in the literature.

² Music Information Retrieval Evaluation eXchange, <http://www.music-ir.org/mirex>, accessed 2015-04-30

³ <https://github.com/DDMAL/salami-data-public/releases/tag/2.0>, accessed 2015-04-30

As explicated in [18], baseline scores can be estimated using variations of regularly or randomly spaced grids as synthetic boundary estimates. For an evaluation tolerance of ± 0.5 seconds, the baseline within our test data set is $F_1 \approx 0.15$. Upper bounds, on the other hand, can be derived from the differences between two independent annotations of the same musical pieces. By analyzing the items within our test data set that have been annotated twice (439 pieces), we calculated $F_1 \approx 0.74$.

In the existing literature, both tolerances of ± 0.5 and ± 3 seconds are commonly used. For this contribution, due to space constraints, we only evaluate for ± 0.5 seconds, where the explorable space, that is, the distance between the lower baseline and the upper bound exhibited in human ground-truth annotations is much greater than for ± 3 seconds (with lower and upper bounds at 0.33 and 0.80, respectively). Our evaluation code is equivalent to the boundary detection implemented in `mir_eval` [13], omitting the borders at the beginning and end of sound files.

Nieto et al. [8] have identified the $F_{0.58}$ measure to be more perceptually informative than the typically used F_1 measure. As this is a relatively new finding and it is not as well established as the F_1 measure (which is, e.g., used in MIREX), we base threshold optimization and model selection on the latter.

4.3 Combination of features

Building on [3], we combine mel-scaled log-magnitude spectrograms (MLS) and self-similarity lag matrices (SSLM) as input features to the CNN. A decomposition of MLS into harmonic and percussive components (feature ‘MLS-HPSS’) and the combination of two SSLMs, one a high-resolution, low lag matrix, the other one a low-resolution, high lag matrix, provides even more structural information to the network. We mainly compare two models: ‘MLS + SSLM-near’ (the model developed in [3]), and the more complex and computationally more expensive model ‘MLS-HPSS + combined SSLM’, integrating all available input features.

The different input features are fused at a relatively late stage in the network (see Figure 2), using a convolutional layer which spans all the vertical (frequency or lag time) components, but only a very short time context. This is motivated by the assumption that the input features are strongly correlated in time. Figure 3 shows boundary recognition scores for the ‘MLS + SSLM-near’ model and three different context widths (1, 3 and 5 bins), evaluated on the validation set. As can be seen, a temporal context for the fusion layer of more than a single bin does not improve the results.

4.4 Consideration of multiple annotations

Up to now, CNN-based boundary recognition algorithms have been trained on data sets with just one annotation version per music piece. SALAMI data contains double annotations for the majority of training examples. It

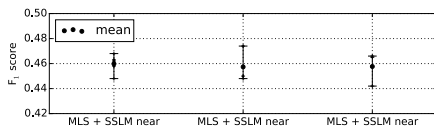


Figure 3: Comparison of boundary recognition F_1 scores for different widths of the CNN fusion layer.

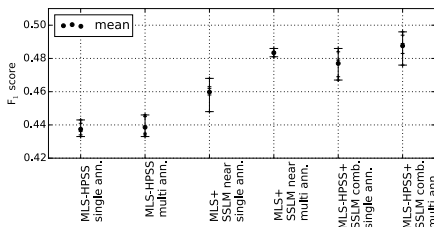


Figure 4: Comparison of boundary recognition F_1 scores for different models trained with single and multiple annotations.

is worth inspecting whether multiple, potentially contradicting annotations help or confuse the CNN training process. Figure 4 shows the results for three different models trained with single and multiple annotations, respectively, evaluated on the validation set. Employing multiple annotations by duplicating audio features and applying the alternative target annotations, the number of training examples increase from 1198707 (with 70317×3 positive examples) to 1670944 (98913×3 positive examples) data points, corresponding to +39%. A positive effect can be observed for models with more versatile structural information available for the network. In these cases, the increase of the F_1 score is in the range of 1–2%.

4.5 Integration of fine annotation

Traditionally, boundary detection in MIR has been performed on only one structural level. As motivated in Section 1, we would like to deal with the ambiguity of annotating boundaries at a certain level of detail by capitalizing on the two-level annotations present in the training data set. This way, the neural network should be able to refine its distinction between main and secondary boundaries.

We explored three different modes for the combination of coarse and fine boundaries: Firstly, by using only one target output vector by assigning full training weights to coarse labels and reduced training weights (e.g., factors of 0.3 or 0.5) to fine labels. Secondly, by using two target outputs with equal weights, one for the coarse labels and one for the fine labels (‘concat’ mode). And finally, using two target outputs, with coarse labels and full weights assigned to the first output vector. Fine labels are assigned to the second output vector, but only where they are distinct from a coarse label (‘contrast’ mode). This should create a more pronounced contrast between coarse and fine labels

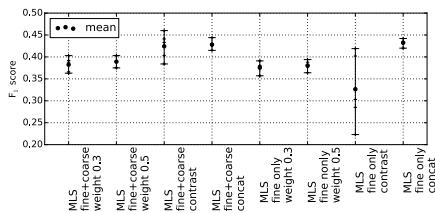


Figure 5: Comparison of boundary recognition F_1 scores for different integration modes of the second-level ‘fine’ annotation, evaluated on our validation set.

with the potential danger of some contradiction.

Not for all of our training data two-level annotations were available. We tried two variations: For the first one, we put coarse boundaries where fine ones were not available (‘fine + coarse’), and for the second one, we used only those annotations with two levels available (‘fine only’), effectively reducing the number of training examples including multiple annotations to 1224891 (with 74400×3 positive examples).

Figure 5 shows the results for the three combination modes (with different weighting parameters) and two data set variations, computed on MLS input features and evaluated on the validation set. The combination modes for coarse and fine data with two output vectors perform better than the ones with only one output. The ‘contrast’ mode exhibits instabilities for the results, most probably due to the relatively small validation data set. We selected the best-performing and reliable ‘concat’ mode with two output units as our working model. The distinction between ‘fine + coarse’ and ‘fine only’ variations is more or less inconclusive, with very little advantage for the latter. However, as the spreading of F_1 scores is less for ‘fine only’, we settled for this variation of the ‘concat’ mode.

5. RESULTS

Figure 6 shows boundary recognition scores (on the primary ‘coarse’ boundaries) of several of our models, with peak-picking thresholds optimized on the validation set, and results evaluated on the test set. Each model variation has been trained and evaluated five times. The individual, mean and ‘bagged’ results are shown in the graph. ‘Bagging’ means that the outputs of all five models are averaged and peak-picking is performed on the result, thereby reducing statistical variations. Using a MLS-HPSS decomposition does not score significantly higher than MLS only. Likewise, using a combination of SSLM ‘near’ (14 seconds lag, high resolution) and ‘far’ (88 seconds lag, low resolution) does not score higher than SSLM ‘near’ only. However, in combination, it can be seen that all ‘MLS-HPSS + combined SSLM’ results are higher than their respective equivalents of ‘MLS + SSLM-near’. For both combined models, using multiple annotations raises the scores relative to single annotations. Additional fine

Algorithm	F_1	$F_{.58}$	Rec.	Prec.
Upper bound (est.)	.74	.74		
<i>All features, multi+fine ann.</i>	.508	.529	.502	.572
<i>MLS+SSLM-near, multi+fine</i>	.496	.506	.509	.536
<i>MLS+SSLM-near, single ann.</i>	.469	.466	.504	.475
SUG1 (2014)	.422	.442	.422	.490
MP2 (2013)	.294	.280	.362	.271
MP1 (2013)	.276	.270	.311	.269
NB1 (2014)	.270	.246	.374	.229
KSP2 (2012)	.263	.231	.422	.209
Baseline (est.)	.15	.21		

Table 1: Boundary recognition scores for coarse boundaries at a tolerance of ± 0.5 seconds, evaluated on our SALAMI 2.0 test dataset. Comparison of our models (in italics) with the five best-performing algorithms of the MIREX campaigns 2012 through 2014.

Algorithm	F_1	$F_{.58}$	Rec.	Prec.
Upper bound (est.)	.75	.76		
<i>All features, multi+fine ann.</i>	.485	.523	.443	.587
<i>MLS+SSLM-near, multi+fine</i>	.478	.515	.439	.576
Baseline (est.)	.23	.17		

Table 2: Boundary recognition scores of two of our models for ‘fine’, second-level boundaries at a tolerance of ± 0.5 seconds, evaluated on our SALAMI 2.0 test dataset.

annotations for CNN training further increase the scores. On the right-hand-side of Figure 6 different feature combinations using multiple fine annotations are shown. The more perspectives on the audio provided as input, the higher the scores.

See Table 1 for a listing of our results in comparison to the best-performing algorithms of the MIREX campaigns 2012 through 2014. All results have been evaluated on SALAMI 2.0 data. Note that the scores are generally lower than for SALAMI 1.2 annotations (cf. [18]). The reason is that in the new data set version many formerly ‘trivial boundaries’ (sitting at the beginning or end of sound files) have been corrected. These boundaries have moved away from the borders and are now headed, or trailed, respectively, by silence or crowd noise, and are therefore more difficult to predict. The ‘MLS+SSLM-near’ model trained with single annotations is equivalent to the model used in [3], with an additional dense layer in the present work. ‘All features’ denotes the ‘MLS-HPSS + combined SSLM’ model, yielding the best boundary prediction results.

Table 2 lists boundary recognition results of the ‘fine’ output unit of our network, trained and evaluated on the ‘small-scale’, second-level annotations of the SALAMI 2.0 data set. To our knowledge, only McFee and Ellis [6] have so far evaluated their algorithms (as well as SMGA [15]) on the secondary boundaries. They report F_1 scores up to 0.292 ± 0.15 on the SALAMI 1.2 dataset.

Table 3 presents boundary recognition results on the Beatles-ISO dataset,⁴ comprised of all 12 Beatles albums with 180 songs in total. We used the best-scoring model from above, using all input features, trained on

⁴ <http://isophonics.net/content/reference-annotations-beatles>, accessed 2015-04-30

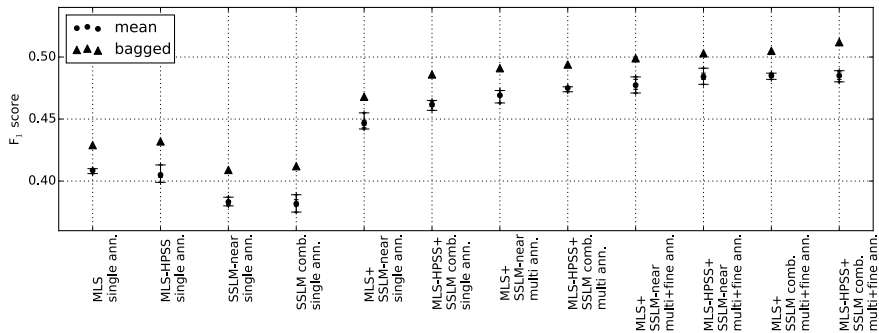


Figure 6: Comparison of boundary recognition F_1 scores on SALAMI 2.0 data for different models under examination. Threshold optimization performed on validation set, evaluation done on test set.

Algorithm	F_1	$F_{.5S}$	Rec.	Prec.
<i>All features, multi+fine ann.</i>	.558	.590	.522	.640
<i>MLS+SSLM+near, multi+fine</i>	.526	.553	.500	.597
SUG1	.424	.457	.385	.510
MP2-beatles	.334	.321	.376	.311
MP2-salami	.322	.313	.355	.309
NB1	.286	.274	.332	.266
MP1	.278	.280	.285	.285
NB2	.266	.255	.302	.247
NB3	.227	.211	.287	.200
Baseline (est.)	.15	.22		

Table 3: Boundary recognition scores at a tolerance of ± 0.5 seconds, evaluated on the Beatles-ISO dataset (180 songs). Two of our models are compared to several published state-of-the-art algorithms.

SALAMI 2.0 with multiple coarse and fine annotations. We were able to compare the predictions of our CNN to the best-performing algorithms of last years’ MIREX submissions by Schlüter et al. (SUG1, personal communication), McFee and Ellis [5] (MP1 and MP2, the latter optimized either for SALAMI and Beatles data),⁵ and Nieto and Bello [7] (NB1, NB2, NB3),⁶ respectively. Note that the scores of our models are above those of other state-of-the-art algorithms by a large margin, although we have not trained or tuned our models in any way specifically on the kind of music realized by the Beatles.

6. DISCUSSION AND OUTLOOK

In this contribution, we have dealt with the prediction of musically relevant structural boundaries, focused primarily on the stylistically mixed SALAMI data set in its latest version 2.0, with additional evaluation on the Beatles-ISO data set.

We have re-used the CNN architecture developed in [3] with some modifications. On the one hand, we have fed it a

⁵<https://github.com/bmcfee/olda>, accessed 2015-05-01

⁶<https://github.com/urinieto/SegmenterMIREX2014>, accessed 2015-05-01

combination of different input features and have been able to show that the CNN is able to produce highest-scoring results with HPSS-decomposed mel-scaled spectrograms (MLS) in combination with self-similarity lag matrices (SSLMs) on two different time-scales, covering both structural detail and longer time context. On the other hand, we have taken advantage of the fact that the SALAMI data set is annotated on two structural levels, and, for the most part, by two independent annotators. The integration of this supplementary data helps the CNN to take better informed decisions between primary and secondary boundaries. Evaluated on SALAMI 2.0 data, we have been able to raise the state of the art from the best MIREX submission [14] at $F_1 = 0.422$, and our previous point of reference [3] at $F_1 = 0.469$ to the score of $F_1 = 0.508$ for the best model, integrating all available input features, as well as multiple and two-level annotations. As the CNN model trained on two-level annotation possesses two output units, its subsequent application also yields two independent predictions for ‘coarse’ and ‘fine’ boundaries.

Although we have not touched (nor listened to) music by the Beatles while developing our models, evaluation on this data set reveals that our models are quite robust, yielding a boundary recognition score of $F_1 = 0.558$, which is significantly higher than the previously published state of the art.

We are still actively exploring the possibilities of CNNs applied to music structure discovery. That said, we have neither exhaustively researched the space of possible input features, nor all meaningful variations of model architecture and learning parameters. There is plenty of remaining headroom to the ‘upper bound’ inter-annotator F_1 scores.

7. ACKNOWLEDGMENTS

This research is funded by the Federal Ministry for Transport, Innovation & Technology (BMVIT) and the Austrian Science Fund (FWF) through project TRP 307-N23 and the Vienna Science and Technology Fund (WWTF) through project MA14-018.

8. REFERENCES

- [1] Derry Fitzgerald. Harmonic/percussive separation using median filtering. In *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx-10)*, Graz, Austria, 2010.
- [2] Jonathan Foote. Automatic audio segmentation using a measure of audio novelty. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME'00)*, volume 1, pages 452–455, New York, USA, 2000.
- [3] Thomas Grill and Jan Schlüter. Music Boundary Detection Using Neural Networks on Spectrograms and Self-Similarity Lag Matrices. In *Proceedings of the 23rd European Signal Processing Conference (EU-SPICO 2015)*, Nice, France, 2015.
- [4] Lie Lu, Muyuan Wang, and Hong-Jiang Zhang. Repeating pattern discovery and structure analysis from acoustic music data. In *MIR '04: Proceedings of the 6th ACM SIGMM international workshop on Multimedia information retrieval*, pages 275–282, New York, USA, 2004.
- [5] Brian McFee and Daniel P. W. Ellis. Learning to segment songs with ordinal linear discriminant analysis. In *International conference on acoustics, speech and signal processing*, ICASSP, 2014.
- [6] Brian McFee and Daniel PW Ellis. Analyzing song structure with spectral clustering. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 405–410, Taipei, Taiwan, 2014.
- [7] Oriol Nieto and Juan Pablo Bello. Music Segment Similarity Using 2D-Fourier Magnitude Coefficients. In *Proceedings of the 39th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 664–668, Florence, Italy, 2014.
- [8] Oriol Nieto, Morwaread M Farbood, Tristan Jehan, and Juan Pablo Bello. Perceptual analysis of the f-measure for evaluating section boundaries in music. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 265–270, Taipei, Taiwan, 2014.
- [9] Jouni Paulus and Anssi Klapuri. Music structure analysis by finding repeated parts. In *AMCMM '06: Proceedings of the 1st ACM workshop on Audio and music computing multimedia*, pages 59–68, New York, USA, 2006.
- [10] Jouni Paulus and Anssi Klapuri. Acoustic features for music piece structure analysis. In *Proceedings of the 11th International Conference on Digital Audio Effects (DAFx-08)*, Espoo, Finland, 2008.
- [11] Jouni Paulus and Anssi Klapuri. Music structure analysis using a probabilistic fitness measure and a greedy search algorithm. *IEEE Transactions on Audio, Speech, and Language Processing*, 17(6):1159–1170, 2009.
- [12] Jouni Paulus, Meinard Müller, and Anssi Klapuri. Audio-based music structure analysis. In *Proceedings of the 11th International Conference on Music Information Retrieval (ISMIR 2010)*, pages 625–636, 2010.
- [13] Colin Raffel, Brian McFee, Eric J Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel PW Ellis. mir_eval: A transparent implementation of common MIR metrics. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, Taipei, Taiwan, 2014.
- [14] Jan Schlüter, Karen Ullrich, and Thomas Grill. Structural segmentation with convolutional neural networks mirex submission. In *Tenth running of the Music Information Retrieval Evaluation eXchange (MIREX 2014)*, 2014.
- [15] Joan Serrà, Meinard Müller, Peter Grosche, and Josep Ll. Arcos. Unsupervised music structure annotation by time series structure features and segment similarity. In *IEEE Transactions on Multimedia*, 16(5):1229–1240, 2014.
- [16] Jordan Bennett Louis Smith, John Ashley Burgoyne, Ichiro Fujinaga, David De Roure, and J Stephen Downie. Design and creation of a large-scale database of structural annotations. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*, pages 555–560, 2011.
- [17] Douglas Turnbull, Gert Lanckriet, Elias Pampalk, and Masataka Goto. A supervised approach for detecting boundaries in music using difference features and boosting. In *Proceedings of the 8th International Conference on Music Information Retrieval (ISMIR 2007)*, pages 51–54, 2007.
- [18] Karen Ullrich, Jan Schlüter, and Thomas Grill. Boundary Detection in Music Structure Analysis using Convolutional Neural Networks. In *Proceedings of the 15th International Society for Music Information Retrieval Conference (ISMIR 2014)*, Taipei, Taiwan, 2014.

NEUROIMAGING METHODS FOR MUSIC INFORMATION RETRIEVAL: CURRENT FINDINGS AND FUTURE PROSPECTS

Blair Kaneshiro

Center for Computer Research in Music and Acoustics
Stanford University, Stanford, CA, USA
blairbo@ccrma.stanford.edu

Jacek P. Dmochowski

Department of Psychology
Stanford University, Stanford, CA, USA
dmochowski@gmail.com

ABSTRACT

Over the past decade and a half, music information retrieval (MIR) has grown into a robust, cross-disciplinary field spanning a variety of research domains. Collaborations between MIR and neuroscience researchers, however, are still rare, and to date only a few studies using approaches from one domain have successfully reached an audience in the other. In this paper, we take an initial step toward bridging these two fields by reviewing studies from the music neuroscience literature, with an emphasis on imaging modalities and analysis techniques that might be of practical interest to the MIR community. We show that certain approaches currently used in a neuroscientific setting align with those used in MIR research, and discuss implications for potential areas of future research. We additionally consider the impact of disparate research objectives between the two fields, and how such a discrepancy may have hindered cross-discipline output thus far. It is hoped that a heightened awareness of this literature will foster interaction and collaboration between MIR and neuroscience researchers, leading to advances in both fields that would not have been achieved independently.

1. INTRODUCTION

Since its inception, music information retrieval (MIR) has been characterized as an interdisciplinary and multifaceted field, drawing from such diverse domains as information science, music, computer science, and audio engineering to explore topics ranging from indexing and retrieval to musical analysis and user studies [22, 24]. The field has become increasingly collaborative over time, and cross-disciplinary output has grown [33].

However, one field that has yet to establish itself as a definitive sub-discipline of MIR is that of neuroscience. Recent papers by Aucouturier and Bigand [6, 7] have highlighted the challenges faced by MIR researchers attempting to publish in cognitive science and neuroscience journals, pointing out that MIR approaches have occupied at

best a marginal or incidental role in that literature. The authors cite as a main obstacle a fundamental lack of interest, or understanding, from the cognitive science/neuroscience community. At the same time, the few brain-based MIR studies published to date [16, 40, 52] have emphasized application over background, potentially leaving readers lacking sufficient introduction to the imaging technique and brain response of interest. As things currently stand, the fields of MIR and neuroscience operate largely independently, despite sharing approaches and questions that might benefit from cross-disciplinary investigation.

In an effort to begin reconciling these two fields, the present authors—whose backgrounds collectively span music, neuroscience, and engineering—present a review of studies drawn from the music neuroscience literature and examine their relevance to MIR research. While such a review will not immediately resolve the significant philosophical issues described above, it may perhaps open a window between the two disciplines by highlighting shared approaches and potential collaborations while acknowledging differences in aims and motivations. Envisioned outcomes are twofold: First, that MIR researchers may find, in brain responses, a new setting to apply analysis techniques already developed for other types of data; and second, and more importantly, that heightened awareness of this literature will increase collaborations between MIR and neuroscience researchers, advancing both fields and leading to the formation of a robust cross-discipline.

Since a review of the entire literature on music and neuroscience would be beyond the scope of this paper, we narrow the present focus to approaches that align closely with MIR applications. For rigor, we include only peer-reviewed papers, though interested readers are encouraged to visit other venues—including but not limited to ICMPC, SMPC, and late-breaking ISMIR proceedings—for a wealth of additional ideas and findings. The primary focus here is on EEG, though behavioral and fMRI studies will be touched upon as appropriate.

The remainder of this paper is structured as follows. First, we evaluate the suitability of various neuroimaging modalities for MIR research (§2). We then review three neuroimaging approaches used in music research (§3) and consider how these methods, and others, might be used for MIR research (§4). We conclude with a discussion of diverging objectives between the two fields, and opportunities for future cross-disciplinary research (§5).



© Blair Kaneshiro, Jacek P. Dmochowski.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Blair Kaneshiro, Jacek P. Dmochowski. "Neuroimaging Methods for Music Information Retrieval: Current Findings and Future Prospects", 16th International Society for Music Information Retrieval Conference, 2015.

2. NEUROIMAGING METHODS FOR MIR

Neuroimaging is the use of magnetic, electrical, hemodynamic, optical, or chemical means to measure activity in the central nervous system, most often the cerebral cortex (Table 1). The central idea behind bridging neuroimaging with MIR is that music is encoded by the brain, and thus can be “read out” or decoded using imaging techniques. In order to exploit this idea, it would be advantageous to track neural activity at the temporal resolution of music (i.e., milliseconds), which necessitates the use of techniques that provide direct electromagnetic measures of neural activity. While techniques measuring hemodynamic responses, such as functional magnetic resonance imaging (fMRI), provide superb spatial resolution that can indirectly probe neural activation on a millimeter scale and elucidate the functional brain networks recruited to process music, the sluggishness of these responses makes them less likely to play a role in MIR.

	EEG	MEG	ECoG	fMRI	DTI
Temporal Resolution	high	high	high	low	NA
Spatial Resolution	low	low	high	high	high
Invasiveness	low	low	high	low	low
Mobility/Portability	high	low	low	low	low
Field of View	large	large	small	large	large
Expense to Operate	low	high	NA	high	high

Table 1. Characteristics of neuroimaging techniques frequently used in music and auditory research. Adapted from Mehta and Parasuraman [39].

On the other hand, electroencephalography (EEG) and magnetoencephalography (MEG) provide millisecond temporal resolution that can in principle be used to infer properties of the stimuli evoking encephalographic responses. EEG and MEG consist of sensors placed at or near the scalp surface that detect mass superpositions of activity in the cerebral cortex. The signal-to-noise ratio (SNR) of EEG/MEG is inherently low, typically on the order of -20 dB. However, as activity is usually collected over a spatial aperture consisting of tens or hundreds of sensors, multivariate approaches can be used to derive spatial filters that will enhance the desired signal while suppressing the noise. The limitation of EEG/MEG is low spatial resolution that results from a spatial smoothing of the evoked signal and renders it difficult to localize the underlying source. In order to achieve fine resolution in both space and time, electrodes can be placed directly on the cortical surface, an invasive practice that is feasible only in the case of neurological disease where it is known as electrocorticography (ECoG), which has been recently employed to study processing of music [47, 48, 55]. Note, however, that in the context of MIR, precise spatial localization is likely not a fundamental requirement. All of the above techniques refer to imaging the function of the brain; methods that measure the connections among brain areas, such as diffusion tensor imaging (DTI), have also been used in the context of music research (e.g., [38]).

In order to feasibly integrate neuroimaging with MIR, a form of imaging that is inexpensive, noninvasive, and

finely temporally resolved is required. For these reasons, our primary focus in the present paper is on EEG, which represents the most promising modality for bridging neural responses with MIR. Moreover, EEG offers a whole-brain field of view that allows for studying the interaction of distributed brain areas during musical processing.

3. APPROACHES OF INTEREST

In this section we review three approaches that may prove useful for MIR. The first is an early-latency response generated by the auditory brainstem, while the latter two involve longer latency cortical responses.

3.1 The Frequency-Following Response

The frequency-following response (FFR) is an early-latency subcortical response generated by the auditory brainstem less than 10 msec after an auditory stimulus occurs. It is a sustained, phase-locked response that oscillates at the same frequency as an auditory stimulus to such an extent that the stimulus can be “played back” from an average of many trials of the brain response [25].

The FFR is typically recorded from a single electrode at the vertex of the head, plus reference and ground electrodes. The response is averaged over many stimulus presentations, and is usually analyzed in the frequency or time-frequency domain. The FFR has an especially low SNR; therefore, FFR experiments require on the order of hundreds or thousands of stimulus presentations. The frequency range of interest for this response is primarily under 1,000 Hz, and studies presented here generally use complex, synthesized stimuli with fundamental frequencies no greater than 300 Hz. An introduction to the response and technique can be found in the 2010 tutorial by Skoe and Kraus [51], and recent findings pertaining to music are summarized in a 2013 review by Bidelman [9].

Despite being an early, low-level auditory response, the FFR has been found to show effects of learning-based neural plasticity. Its involvement in the music literature grew out of speech studies that compared subcortical responses of speakers of tone languages, such as Mandarin and Thai, to those of English speakers. These studies showed that FFRs to certain pitch-varying phonemes and phoneme-like stimuli were more robust in the tone-language speakers than in the English speakers, pointing to experience-dependent processing enhancements [29–32, 56]. Trained musicians, who possess a complementary type of pitch expertise, became a population of interest in generalizing these findings. For example, a study by Wong et al. [62] showed that musicians exhibited more robust encoding of Mandarin phonemes than did nonmusicians, despite not being tone-language speakers.

The first study to investigate the FFR specifically in response to musical stimuli was a 2007 study by Musacchia and colleagues [41]. Here, musicians’ enhanced subcortical encoding of speech and musical stimuli presented in audio, visual, and audiovisual modalities could be identified in both the time and frequency domains of the brain

response. Subsequent studies have investigated encoding of musical intervals by musicians and nonmusicians [34], as well as encoding of music by both musicians and Mandarin speakers [10, 11].

Musical characteristics of the stimuli have also been found to modulate the strength of the FFR. A 2009 study by Bidelman and Krishnan [12], revealed enhanced encoding for consonant versus dissonant musical intervals. The authors later found a similar effect in responses to pleasant (major/minor) versus unpleasant (augmented/diminished) triads [13]. It should be noted that these results cannot be merely a reflection of the acoustical properties of the stimuli, as the consonant and dissonant intervals are interleaved (e.g., the dissonant tritone lies between the consonant P4 and P5), as are the constituent intervals (major and minor thirds) comprising the different types of musical triads.

3.2 Single-Trial EEG Classification

We now move from the auditory brainstem to the cerebral cortex, where responses begin roughly 50 msec after stimulus onset and are typically recorded from between 32–256 electrodes arranged across the surface of the scalp at regular intervals, often by means of a cap or net. Cortical responses are generally analyzed in a lower frequency range than FFRs, usually below 50 or 60 Hz.

Cortical EEG research has a long history of univariate analysis. Readers may be familiar with time-averaged event-related potential (ERP) studies, which focus on amplitudes and latencies of particular waveform peaks from selected electrodes. Some recent studies have taken a different approach to EEG analysis by classifying single trials of the brain response. The goal in this case is to correctly predict, from the brain response, which stimulus the participant was experiencing (see Blankertz et al. [15] for an introduction and tutorial). This multivariate approach enables data from multiple electrodes and time points to be analyzed at once. Classification of neuroimaging data has a longer history in fMRI (as multi-voxel pattern analysis [43]) than in EEG; however, the overarching methodology lends itself well to extracting stimulus- or task-relevant components out of noisy, high-dimensional EEG data, as is done with other types of data used in music research [50].

The first single-trial EEG classification study focusing on musical stimuli was published in 2011 by Schaefer and colleagues [49]. They found that brain responses to seven short excerpts of naturalistic music¹ from a variety of genres could be classified significantly above chance. More recently, Stober et al. recorded EEG responses from East African listeners who heard twelve Western and twelve East African rhythms, and used deep-learning techniques to predict both the rhythm family of a stimulus (2-class problem) as well as the individual rhythm (24-class problem) from the EEG [52]. The prediction task of EEG classification has also extended beyond characterizing the stimuli to labeling listeners' emotional states—for example, in response to music videos [28] and musical excerpts [16].

¹ The term “naturalistic music” is used to refer to ecologically valid musical material as opposed to controlled, synthesized stimuli.

A brain-computer interface (BCI) is often cited as a general application of single-trial EEG classification [14]. In a musical context, a successful BCI would enable a user to communicate mentally by selectively interacting with an ongoing musical stimulus. Studies by Vlek and colleagues showed that subjective (mentally imposed) metrical accents on a beat sequence could be detected in the EEG response [60], and that a classifier trained upon responses to perceived accents could be used to detect the imagined accents [61]. In a recent EEG study by Treder et al. [58], also working toward BCI application, listeners were played polyphonic musical stimuli wherein each stream produced intermittent “oddball” musical events, and attended to just one of the streams. The authors leveraged the fact that the brain responds differently to attended oddball auditory stimuli than to unattended oddballs, and classified brain responses to just the oddball events in the music in order to identify the attended stream.

3.3 Tracking Temporal Dynamics of Acoustical Features

Certain music cognition studies have drawn explicitly from MIR techniques, utilizing acoustical features developed specifically for music analysis [59]. These studies use short-term (e.g., spectral flux, spectral centroid) and long-term (e.g., musical mode, pulse clarity) acoustical features, computationally extracted from musical stimuli, as a basis for quantitatively comparing stimuli with responses.

A 2010 behavioral study by Alluri and Toiviainen [1] set the foundation for this approach in the music cognition literature. The authors formulated perceptual scales suitable for assessing timbre of naturalistic music, and then linked human ratings of short musical excerpts to the excerpts' constituent short-term acoustical features. Subsequent fMRI studies used a refined set of short-term features, as well as long-term features, to characterize their musical stimuli. Alluri and colleagues identified brain regions whose fMRI time series correlated with those of the acoustical features of a tango piece [2], and later predicted brain activations from the features of a variety of musical excerpts [3]. A 2014 study by Toiviainen and colleagues took the inverse approach, predicting acoustical features from fMRI-recorded responses to Beatles songs [57].

Acoustical feature representation has also been studied in ongoing EEG. In contrast to relatively short epochs used in FFR and classification analysis, ongoing-EEG epochs can span many minutes, and are thus well suited to the analysis of responses to longer musical excerpts such as songs [17]. A 2013 study by Cong and colleagues used the same stimulus and long-term acoustical features as the 2012 Alluri study [2] in an ongoing-EEG paradigm, decomposing the EEG response into temporally independent sources using Independent Component Analysis (ICA), and then identifying sources whose frequency content corresponded to the time courses of the acoustical features [17]. More recently, Lin and colleagues also used EEG ICA sources to link ongoing-EEG responses to musical mode and tempo in shorter musical excerpts [36].

4. MIR APPLICATIONS

In the previous section, we reviewed three approaches used to study brain responses to music: The FFR, which directly encodes the pitch of an auditory stimulus, and two analysis techniques used for classifying and characterizing cortical responses. We will now discuss MIR applications of neuroimaging data. We consider the relevance of each approach to MIR research and assess the added value of analyzing the brain response—over analyzing, for example, the auditory stimulus directly.

4.1 Transcription

The FFR is unique among the auditory responses presented here in that it directly reflects the stimulus. As described above, the FFR has been used primarily as a measure of encoding. To date, its robustness has been the main attribute of interest, reflecting effects of expertise (tone-language speaker or musician) and stimulus properties (musical consonance or pleasantness) in the brain response.

The FFR could prove to be a powerful transcription tool; to our knowledge, this application has not yet been explored. From an MIR perspective, there would be little added value in transcribing responses to the simple musical stimuli used in the FFR studies described here (mostly monophonic, sometimes intervals or triads—see §3.1), as transcription could be easily accomplished directly from the audio. However, selective attention has been found to enhance FFR amplitudes for simultaneously presented speech stimuli [26, 35]; therefore, future research could study this topic further using musical stimuli, for example to extract a melody from polyphonic music—an open topic in audio MIR research, but something a human can accomplish effortlessly. Though FFRs to imagined sounds have yet to be confirmed, an FFR-based transcription system of this kind would certainly open another exciting and novel avenue for future research.

As described above, FFR studies typically involve up to thousands of stimulus repetitions due to low SNR. Therefore, signal-processing techniques that could efficiently extract the FFR out of the EEG—perhaps by recording the response from a montage of multiple electrodes, analogous to the use of multiple microphones in a source-separation scenario—would provide a useful resource for more flexible experiment design, and provide a critical step toward FFR-based transcription.

4.2 Tagging and Annotation

Characterizing musical attributes and listener responses is a recurring goal in MIR research, and has also been explored in EEG research [28, 37, 40]. In their 2010 paper, Alluri and Toivaiainen [1] draw explicit connections between their proposed approach and the use of acoustical features in computational systems for music categorization. Along these lines, the acoustical feature following approach used in neuroimaging studies could extend beyond the prediction of the features from the brain response (as in [57]), toward a global prediction of musical genre

from combinations of these features over time, as is done in audio-based genre classification.

Interestingly, a fine-grained temporal representation of acoustical musical features in the brain response has yet to be explored using noninvasive imaging techniques. While short-term acoustical features were used in the behavioral and fMRI studies discussed above (§3.3), they were averaged or downsampled to match the length of the behavioral stimuli (1.5 seconds) or the sampling rate of fMRI (0.45–0.5 Hz) [1–3, 57]. At the same time, the studies using EEG—arguably the best modality for investigating representation of short-term acoustical features—considered only long-term acoustical features in their analysis [17, 36]. It may be the case, too, that neurally encoded features of music do not correspond exactly to the hand-crafted acoustical features discussed here; therefore, feature-learning approaches could also prove useful for connecting temporally resolved stimulus features to the brain response, whether to study feature processing and representation, or to develop an annotation tool.

Single-trial EEG classification could also be applied to this problem. Of the classification studies discussed here, only one used naturalistic music as stimuli [49]; the others used rhythmic patterns [52, 60, 61] or short events segmented from an ongoing stimulus [58]. One possibility for future MIR application could be to classify responses to a larger set of naturalistic musical excerpts to build, for example, a classification model that surpasses excerpt-level specificity and instead predicts genre, mood, or other global attributes from responses to new musical excerpts.

4.3 Predicting Large-Scale Audience Preferences

Brain responses can also be used to model listener preferences. This topic has been explored to some extent in the music neuroscience literature (e.g., [4]). However, to accomplish a widespread application of this goal—for example, in a neuromarketing setting [5]—would require that responses of the experimental sample generalize beyond that sample to a large-scale measure of success, such as sales of a product or ratings collected from the general public [8].

Recent studies have successfully used brain responses from a small sample to predict large-scale audience preferences. In a 2012 study, Berns and Moore collected fMRI responses and subjective ratings from participants who listened to a set of unknown songs. The authors then tracked the sales of the songs over the next three years and found a brain region whose activity correlated significantly with eventual song popularity [8]. Recent studies by Falk et al. [23] and Dmochowski et al. [21] showed that large-scale success of television commercials could be predicted from fMRI and EEG responses, respectively. In all three of these studies, the brain responses of the experimental sample correlated more strongly with large-scale measures of popularity and success than they did with self-reported preferences of that same sample. These findings lend credence to the theory that brain responses provide objective measures of preference, and that generalizations may be drawn from these responses with greater validity than sub-

Company	Product	Application	Website	Features
Emotiv	EPOC	commercial	http://emotiv.com/	fixed montage, wireless, iOS, Android
NeuroSky	MindWave, MindSet	commercial	http://neurosky.com/	fixed montage, wireless, iOS, Android
EGI	Avatar	research	http://avatareeg.com/	flexible montage and sensors, wireless, Android
Grass	Comet	research	http://www.grasstechnologies.com/	flexible montage and sensors
Neuroelectrics	StarStim	research	http://www.neuroelectrics.com/	flexible montage and sensors, wireless, stimulation

Table 2. Selected portable/mobile EEG systems.

jective ratings from a small experimental sample—even the very simple providing the brain responses. Therefore, MIR researchers may find brain-based measures of preference or success to be a useful channel of information in predicting or modeling large-scale music popularity.

4.4 Portable/Mobile EEG

While not an application per se, another area of growing interest in neuroscience involves portable and mobile EEG systems. It should be noted that nearly all of the studies reviewed here were conducted in controlled laboratory settings; thus, the listening experiences of the experimental participants likely did not reflect their experiences of music in everyday life. However, a number of commercial- and research-grade systems have come to market over the past decade (Table 2), and have recently begun to gain traction in the scientific literature as valid data-acquisition tools.

In an MIR context, a 2013 study by Morita et al. used the NeuroSky MindSet to assess mental states in response to music [40], and the 2014 study by Stober and colleagues (§3.2) used a portable Grass system for data collection [52]. Other recent scientific publications report real-time 3D imaging implementations using wireless EEG with a smartphone interface built using Emotiv equipment [44, 54], and a 2014 study by De Vos and colleagues showed that usable single-trial auditory responses could be recorded from a custom portable apparatus, also built off of the Emotiv system [18, 19]. The adoption of such methodologies by the scientific community presents an opportunity for MIR researchers to study music consumption and music processing in real-world listening situations [36].

5. DISCUSSION

In this review, we have surveyed neuroimaging techniques that can be used in MIR research, and highlighted a number of potential research topics spanning the two fields. Why, then, have collaborations not flourished to date?

One answer may emerge from a consideration of fundamental motivational differences between the two fields. Neuroscience, by definition, is the study of the brain; therefore, the thrust of much neuroscientific research is to gain an understanding of brain functioning underlying processing of various stimuli, including music. As a result, experiment design, data analysis, and interpretation of results will tend toward this goal, even when analysis involves decoding or prediction of stimulus or response features. A useful perspective on this topic is provided by Naselaris and colleagues [42], who characterize encoding versus decoding approaches used in fMRI research: Encoding ap-

proaches assess variations in neural space in response to variations in stimulus space, or perhaps seek to predict the brain response from the stimulus. Decoding, on the other hand, seeks to predict information about the stimulus from the brain response. In a neuroscientific setting, both approaches are used to map stimulus features to responses in order to better understand brain processing.

This objective is clearly evident in the studies reviewed above (§3). The FFR, providing arguably the most decodable brain signal, is used primarily to study neural encoding of auditory stimuli. One outcome of single-trial classification is the identification of temporal and spatial EEG components that best discriminate or differentiate stimuli or stimulus categories. The acoustical feature studies also focused upon identifying brain areas whose activity covaried with the stimuli, and not specifically on transcription. Of the approaches described above, perhaps only the BCI-focused EEG classification studies are purely application-based, with system performance taking priority over an exploration of the underlying neural processing—though an understanding of the latter is often a design consideration in the development of a high-performing BCI system.

MIR, on the other hand, tends to be a more application- and goal-oriented field [7]. For MIR researchers, then, brain data may serve more as a medium through which information about music may be recovered, than as the fundamental object of investigation. This disparity in what the brain, and brain data, represent in the overall goal of the research may be partly responsible for the lack of connection and collaboration between the two fields to date.

Another likely hindrance to the incorporation of neuroscientific techniques in MIR is access to data. Historically, researchers have had to acquire their own data, which requires access to equipment as well as domain-specific expertise in experiment design and data collection. Following that, data preprocessing and analysis can require significant signal-processing proficiency to extract stimulus-related information from noisy EEG recordings, especially for the single-trial and ongoing-EEG approaches discussed above. Luckily, the global scale of music neuroscience research now underway should provide many opportunities for collaboration, whereby MIR researchers may bypass some of the above steps if they wish. In addition, the creation of publicly available repositories of neuroimaging data has become a recent area of focus in the fMRI community [45, 46], and the EEG community is following suit (music-related EEG datasets include Koelstra et al.'s DEAP [27] and Stober et al.'s OpenMIIR [53]). Such public datasets, as well as open-source analysis packages such as EEGLAB [20], can facilitate cross-disciplinary research

even in the absence of formal collaborations.

While the fields of MIR and neuroscience have yet to form a strong connection, there exist many opportunities for collaboration that could advance both fields. It is hoped that the studies and ideas presented in this review will prove useful to both MIR researchers and neuroscientists. It is likely that the two fields will take some time to grow closer; therefore, MIR output using neuroscientific data may not immediately reach the neuroscientific audience (nor should it be intended to). Even so, we hope that a greater knowledge of neuroscientific approaches and findings will spark the interest of MIR researchers and lead to future intersections between these two exciting fields.

6. ACKNOWLEDGMENTS

This research is supported by the Wallenberg Network Initiative: Culture, Brain, Learning. The authors thank Jonathan Berger, Anthony Norcia, Jorge Herrera, and four anonymous ISMIR reviewers for their helpful suggestions and feedback on this paper.

7. REFERENCES

- [1] V. Alluri and P. Toivaiainen. Exploring perceptual and acoustic correlates of polyphonic timbre. *Music Perception: An Interdisciplinary Journal*, 27(3):223–242, 2010.
- [2] V. Alluri, P. Toivaiainen, I. P. Jääskeläinen, E. Gleeran, M. Sams, and E. Brattico. Large-scale brain networks emerge from dynamic processing of musical timbre, key and rhythm. *NeuroImage*, 59(4):3677–3689, 2012.
- [3] V. Alluri, P. Toivaiainen, T. E. Lund, M. Wallentin, P. Vuust, A. K. Nandi, T. Ristaniemi, and E. Brattico. From Vivaldi to Beatles and back: Predicting lateralized brain responses to music. *NeuroImage*, 83(0):627–636, 2013.
- [4] E. Altenmüller, K. Schürmann, V. K. Lim, and D. Parlitz. Hits to the left, flops to the right: Different emotions during listening to music are reflected in cortical lateralisation patterns. *Neuropsychologia*, 40(13):2242–2256, 2002.
- [5] D. Ariely and G. S. Berns. Neuromarketing: The hope and hype of neuroimaging in business. *Nature Reviews Neuroscience*, 11(4):284–292, 2010.
- [6] J. J. Aucouturier and E. Bigand. Mel Cepstrum & Ann Ova: The difficult dialog between MIR and music cognition. In *ISMIR*, pages 397–402, 2012.
- [7] J. J. Aucouturier and E. Bigand. Seven problems that keep MIR from attracting the interest of cognition and neuroscience. *Journal of Intelligent Information Systems*, 41(3):483–497, 2013.
- [8] G. S. Berns and S. E. Moore. A neural predictor of cultural popularity. *Journal of Consumer Psychology*, 22(1):154–160, 2012.
- [9] G. M. Bidelman. The role of the auditory brainstem in processing musically relevant pitch. *Frontiers in psychology*, 4:264, 2013.
- [10] G. M. Bidelman, J. T. Gandour, and A. Krishnan. Cross-domain effects of music and language experience on the representation of pitch in the human auditory brainstem. *Journal of cognitive neuroscience*, 23(2):425–434, February 2011.
- [11] G. M. Bidelman, J. T. Gandour, and A. Krishnan. Musicians and tone-language speakers share enhanced brainstem encoding but not perceptual benefits for musical pitch. *Brain and cognition*, 77(1):1–10, October 2011.
- [12] G. M. Bidelman and A. Krishnan. Neural correlates of consonance, dissonance, and the hierarchy of musical pitch in the human brainstem. *The Journal of Neuroscience*, 29(42):13165–13171, 2009.
- [13] G. M. Bidelman and A. Krishnan. Brainstem correlates of behavioral and compositional preferences of musical harmony. *Neuroreport*, 22(5):212–216, March 2011.
- [14] B. Blankertz, G. Curio, and K. R. Müller. Classifying single trial EEG: Towards brain computer interfacing. In *Advances in Neural Information Processing Systems*, pages 157–164, 2002.
- [15] B. Blankertz, S. Lemm, M. Treder, S. Haufe, and K. R. Müller. Single-trial analysis and classification of ERP components—a tutorial. *NeuroImage*, 56(2):814–825, 2011.
- [16] R. Cabredo, R. S. Legaspi, P. S. Inventado, and M. Numao. An emotion model for music using brain waves. In *ISMIR*, pages 265–270, 2012.
- [17] F. Cong, V. Alluri, A. K. Nandi, P. Toivaiainen, R. Fa, B. Abu-Jamous, L. Gong, B. G. W. Craenen, H. Poikonen, M. Huotilainen, and T. Ristaniemi. Linking brain responses to naturalistic music through analysis of ongoing EEG and stimulus features. *IEEE Transactions on Multimedia*, 15(5):1060–1069, August 2013.
- [18] M. De Vos, K. Gandras, and S. Debener. Towards a truly mobile auditory braincomputer interface: Exploring the P300 to take away. *International Journal of Psychophysiology*, 91(1):46–53, 2014.
- [19] S. Debener, F. Minow, R. Emkes, K. Gandras, and M. De Vos. How about taking a low-cost, small, and wireless EEG for a walk? *Psychophysiology*, 49(11):1617–1621, 2012.
- [20] A. Delorme and S. Makeig. EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, 134(1):9–21, 2004.
- [21] J. P. Dmochowski, M. A. Bezdek, B. P. Abelson, J. S. Johnson, E. H. Schumacher, and L. C. Parra. Audience preferences are predicted by temporal reliability of neural processing. *Nature communications*, 5:4567, 2014.
- [22] J. S. Downie. Music information retrieval. *Annual Review of Information Science and Technology*, 37(1):295–340, 2003.
- [23] E. B. Falk, E. T. Berkman, and M. D. Lieberman. From neural responses to population behavior: Neural focus group predicts population-level media effects. *Psychological Science*, 23(5):439–445, 2012.
- [24] J. Furlle and J. S. Downie. Interdisciplinary communities and research issues in music information retrieval. In *ISMIR*, pages 215–221, 2002.
- [25] G. C. Galbraith, P. W. Arbagey, R. Branski, N. Comerici, and P. M. Rector. Intelligible speech encoded in the human brain stem frequency-following response. *Neuroreport*, 6(17):2363–2367, November 1995.
- [26] G. C. Galbraith, S. M. Bhuta, A. K. Choate, J. M. Kitahara, and T. A. Mullen. Brain stem frequency-following response to dichotic vowels during attention. *Neuroreport*, 9(8):1889–1893, June 1998.
- [27] S. Koelstra, C. Muhl, M. Soleymani, J. S. Lee, A. Yazdani, T. Pun, T. Ebrahimi, T. Pun, A. Nijholt, and I. Patras. DEAP: A database for emotion analysis using physiological signals. *IEEE Transactions on Affective Computing*, 3(1):18–31, January 2012.
- [28] S. Koelstra, A. Yazdani, M. Soleymani, C. Mhl, J. S. Lee, A. Nijholt, T. Pun, T. Ebrahimi, and I. Patras. Single trial classification of EEG and peripheral physiological signals for recognition of emotions induced by music videos. In *Brain Informatics*, volume 6334 of *Lecture Notes in Computer Science*, pages 89–100. Springer Berlin Heidelberg, 2010.

- [29] A. Krishnan, G. M. Bidelman, and J. T. Gandour. Neural representation of pitch salience in the human brainstem revealed by psychophysical and electrophysiological indices. *Hearing Research*, 268(12):60–66, 2010.
- [30] A. Krishnan, J. T. Gandour, and G. M. Bidelman. The effects of tone language experience on pitch processing in the brainstem. *Journal of NeuroLinguistics*, 23(1):81–95, 2010.
- [31] A. Krishnan, J. T. Gandour, G. M. Bidelman, and J. Swaminathan. Experience-dependent neural representation of dynamic pitch in the brainstem. *Neuroreport*, 20(4):408–413, March 2009.
- [32] A. Krishnan, Y. Xu, J. Gandour, and P. Cariani. Encoding of pitch in the human brainstem is sensitive to language experience. *Cognitive Brain Research*, 25(1):161–168, 2005.
- [33] J. H. Lee, M. C. Jones, and J. S. Downie. An analysis of ISMIR proceedings: Patterns of authorship, topic, and citation. In *ISMIR*, pages 58–62, 2009.
- [34] K. M. Lee, E. Skoe, N. Kraus, and R. Ashley. Selective subcortical enhancement of musical intervals in musicians. *The Journal of Neuroscience*, 29(18):5832–5840, 2009.
- [35] A. Lehmann and M. Schönwiesner. Selective attention modulates human auditory brainstem responses: Relative contributions of frequency and spatial cues. *PLoS one*, 9(1):e85442, 2014.
- [36] Y. P. Lin, J. R. Duann, W. Feng, J. H. Chen, and T. P. Jung. Revealing spatio-spectral electroencephalographic dynamics of musical mode and tempo perception by independent component analysis. *Journal of NeuroEngineering and Rehabilitation*, 11(1), 2014.
- [37] Y. P. Lin, C. H. Wang, T. P. Jung, T. L. Wu, S. K. Jeng, J. R. Duann, and J. H. Chen. EEG-based emotion recognition in music listening. *IEEE Transactions on Biomedical Engineering*, 57(7):1798–1806, July 2010.
- [38] P. Loui, D. Alsop, and G. Schlaug. Tone deafness: A new disconnection syndrome? *The Journal of Neuroscience*, 29(33):10215–10220, 2009.
- [39] R. K. Mehta and R. Parasuraman. Neuroergonomics: A review of applications to physical and cognitive work. *Frontiers in Human Neuroscience*, 7(889), 2013.
- [40] Y. Morita, H. H. Huang, and K. Kawagoe. Towards music information retrieval driven by EEG signals: Architecture and preliminary experiments. In *IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS)*, pages 213–217, June 2013.
- [41] G. Musacchia, M. Sams, E. Skoe, and N. Kraus. Musicians have enhanced subcortical auditory and audiovisual processing of speech and music. *Proceedings of the National Academy of Sciences*, 104(40):15894–15898, 2007.
- [42] T. Naselaris, K. N. Kay, S. Nishimoto, and J. L. Gallant. Encoding and decoding in fMRI. *NeuroImage*, 56(2):400–410, 2011.
- [43] K. A. Norman, S. M. Polyn, G. J. Detre, and J. V. Haxby. Beyond mind-reading: Multi-voxel pattern analysis of fMRI data. *Trends in Cognitive Sciences*, 10(9):424–430, 2006.
- [44] M. K. Petersen, C. Stahllut, A. Stopczynski, J. E. Larsen, and L. K. Hansen. Smartphones get emotional: Mind reading images and reconstructing the neural sources. In *Affective Computing and Intelligent Interaction*, volume 6975 of *Lecture Notes in Computer Science*, pages 578–587. Springer Berlin Heidelberg, 2011.
- [45] R. A. Poldrack, D. M. Barch, J. Mitchell, T. Wager, A. D. Wagner, J. T. Devlin, C. Cumba, O. Koyejo, and M. Milham. Toward open sharing of task-based fMRI data: The OpenfMRI project. *Frontiers in Neuroinformatics*, 7(12), 2013.
- [46] R. A. Poldrack and K. J. Gorgolewski. Making big data open: Data sharing in neuroimaging. *Nature neuroscience*, 17(11):1510–1517, November 2014.
- [47] C. Potes, P. Brunner, A. Gunduz, R. T. Knight, and G. Schalk. Spatial and temporal relationships of electrocorticographic alpha and gamma activity during auditory processing. *NeuroImage*, 97:188–195, 2014.
- [48] C. Potes, A. Gunduz, P. Brunner, and G. Schalk. Dynamics of electrocorticographic (ECoG) activity in human temporal and frontal cortical areas during music listening. *NeuroImage*, 61(4):841–848, 2012.
- [49] R. S. Schaefer, J. Farquhar, Y. Blokland, M. Sadakata, and P. Desain. Name that tune: Decoding music from the listening brain. *NeuroImage*, 56(2):843–849, 2011.
- [50] R. S. Schaefer, S. Furuya, L. M. Smith, B. B. Kaneshiro, and P. Toivaiainen. Probing neural mechanisms of music perception, cognition, and performance using multivariate decoding. *Psychomusicology: Music, Mind, and Brain*, 22(2):168–174, 2012.
- [51] E. Skoe and N. Kraus. Auditory brain stem response to complex sounds: A tutorial. *Ear and hearing*, 31(3):302–324, June 2010.
- [52] S. Stober, D. J. Cameron, and J. A. Grahm. Classifying EEG recordings of rhythm perception. In *ISMIR*, pages 649–654, 2014.
- [53] S. Stober, A. Stermin, A. M. Owen, and J. A. Grahm. Towards music imagery information retrieval: Introducing the OpenMIR dataset of EEG recordings from music perception and imagination. In *ISMIR*, 2015.
- [54] A. Stopczynski, C. Stahllut, J. E. Larsen, M. K. Petersen, and L. K. Hansen. The smartphone brain scanner: A portable real-time neuroimaging system. *PLoS one*, 9(2):e86733, 2014.
- [55] I. Sturm, B. Blankertz, C. Potes, G. Schalk, and G. Curio. ECoG high gamma activity reveals distinct cortical representations of lyrics passages, harmonic and timbre-related changes in a rock song. *Frontiers in Human Neuroscience*, 8(798), 2014.
- [56] J. Swaminathan, A. Krishnan, and J. T. Gandour. Pitch encoding in speech and nonspeech contexts in the human auditory brainstem. *Neuroreport*, 19(11):1163–1167, July 2008.
- [57] P. Toivaiainen, V. Alluri, E. Brattico, M. Wallentin, and P. Vuust. Capturing the musical brain with Lasso: Dynamic decoding of musical features from fMRI data. *NeuroImage*, 88(0):170–180, 2014.
- [58] M. S. Treder, H. Purwins, D. Miklody, I. Sturm, and B. Blankertz. Decoding auditory attention to instruments in polyphonic music using single-trial EEG classification. *Journal of neural engineering*, 11(2):026009, April 2014.
- [59] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, July 2002.
- [60] R. J. Vlek, R. S. Schaefer, C. C. A. M. Gielen, J. D. R. Farquhar, and P. Desain. Sequenced subjective accents for brain-computer interfaces. *Journal of neural engineering*, 8(3):036002, June 2011.
- [61] R. J. Vlek, R. S. Schaefer, C. C. A. M. Gielen, J. D. R. Farquhar, and P. Desain. Shared mechanisms in perception and imagery of auditory accents. *Clinical Neurophysiology*, 122(8):1526–1532, 2011.
- [62] P. C. M. Wong, E. Skoe, N. M. Russo, T. Dees, and N. Kraus. Musical experience shapes human brainstem encoding of linguistic pitch patterns. *Nature neuroscience*, 10(4):420–422, April 2007.

Oral Session 5

Similarity

IMPROVING VISUALIZATION OF HIGH-DIMENSIONAL MUSIC SIMILARITY SPACES

Arthur Flexer

Austrian Research Institute for Artificial Intelligence

arthur.flexer@ofai.at

ABSTRACT

Visualizations of music databases are a popular form of interface allowing intuitive exploration of music catalogs. They are often based on lower dimensional projections of high dimensional music similarity spaces. Such similarity spaces have already been shown to be negatively impacted by so-called hubs and anti-hubs. These are points that appear very close or very far to many other data points due to a problem of measuring distances in high-dimensional spaces. We present an empirical study on how this phenomenon impacts three popular approaches to compute two-dimensional visualizations of music databases. We also show how the negative impact of hubs and anti-hubs can be reduced by re-scaling the high dimensional spaces before low dimensional projection.

1. INTRODUCTION

Visualization via low dimensional projections is one way to produce interfaces that allow navigation and access to music data sets. A very popular and influential approach is the islands-of-music metaphor [14], where representations of similar music form islands on a two-dimensional display. Numerous variations of this approach have been published within the music information retrieval (MIR) community (see e.g. [5, 9, 16, 24]). A recent trend towards more holistic MIR approaches [18, 23] including human computer interaction aspects is likely to increase interest in visualization in the near future. State-of-the-art visualization algorithms are said to be able to visualize high-dimensional data [28]. Precisely for such high-dimensional data a new aspect of the curse of dimensionality, the so called hubness, has been discovered and described within the MIR community [1, 8]. This paper investigates the impact of hubness on visualization of high-dimensional music similarity spaces. In an empirical evaluation of three methods for dimensionality reduction the negative impact of hubness is explored and it is shown how re-scaling of the similarity spaces as a preprocessing step can greatly improve the visualizations.

2. RELATED WORK


Hubness is a general problem of learning in high-dimensional space which has been discovered in MIR [1], but then gained attention in a general machine learning context where it has been discussed as a new aspect of the curse of dimensionality [15, 20]. Hub objects appear very close to many other data objects and anti-hubs very far from most other data objects. The effect is related to the phenomenon of concentration of distances and has been shown to have a negative impact on many tasks including classification [15], nearest neighbor based recommendation [3] and retrieval [21], outlier detection [15] and clustering [19, 26].

Visualization of music similarity spaces via low dimensional projections has a long tradition within MIR. Starting from the influential islands-of-music approach [14, 16], numerous extensions and variations have been developed (see e.g. [5, 9, 24]). Although different methods for dimensionality reduction have been explored, the most popular approach seems to be self-organizing maps [10]. Despite the popularity of these interfaces based on lower dimensional projections, it has not yet been clarified how hubness influences these visualizations. To the best of our knowledge, there is only a single publication concerned with the impact of hubness on visualization [6]. In an analysis of dimensionality reduction of three audio databases to two dimensions using multidimensional scaling, the authors show that projected data tends to be concentrated in a single large cluster centered around the largest hub.

It is important to note that simple dimensionality reduction does not reduce hubness. On the contrary it has been shown that only projections to very few dimensions, well below the intrinsic dimensionality of a data set, are able to reduce hubness, but at the cost of a loss of distance information [15]. On the other hand, results on re-scaling methods to reduce hubness [20] show that it is possible to decrease hubness without changing the intrinsic dimensionality and therefore the information content of the data. Thus a good approach to visualization of high dimensional data might be to first re-scale to reduce hubness without changing the intrinsic dimensionality, and then to apply dimensionality reduction to the re-scaled data.

3. DATA

For our experiments we used two standard music databases: the "GTZAN" collection consisting of $N = 1000$ audio tracks (each 30 s length) evenly spread over ten music gen-

 © Arthur Flexer.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Arthur Flexer. "Improving visualization of high-dimensional music similarity spaces", 16th International Society for Music Information Retrieval Conference, 2015.

res [27]; the “ISMIR2004”¹ collection containing $N = 1458$ tracks of six genres, with full-length audio being available and exhibiting a highly imbalanced genre distribution with classical music comprising almost half of the tracks.

We decided to compute timbre information from the audio, since this is an integral part of many MIR systems and at the same time has already been shown to be susceptible to hubness [3]. Every track is divided into overlapping frames for which 20 MFCCs are being computed which are modeled via a single Gaussian with full covariance matrix. To compute a distance value between two Gaussians the symmetrized Kullback-Leibler (SKL) divergence is used [11]. This results in $N \times N$ distance matrices D_I and D_G for the ISMIR and GTZAN data sets. Please note that SKL is symmetric and non-negative, but does not fulfill the triangle inequality and therefore is not a full metric.

4. METHODS

In what follows we present three methods for dimensionality reduction (TSNE, SAMMON, SOM) and two methods to re-scale distance matrices in order to reduce hubness (MP, SNN). In Section 5 we will use MP and SNN as a preprocessing step before dimensionality reduction. This gives nine different combinations of methods to compare: TSNE, MP TSNE, SNN TSNE, SOM, MP SOM, SNN SOM, SAMMON, MP SAMMON, SNN SAMMON. But first we present evaluation indices that will be used to measure the performance achieved in original and re-scaled data spaces.

4.1 Evaluation measures

Hubness (S^n): To characterize the strength of the hubness phenomenon in a data set we use the so called hubness measure [15]. This is based on the n -occurrences of points x , which is the number of times x occurs in the n -nearest neighbor lists of all other objects in the collection. Hubness is then defined as the skewness of the distribution of n -occurrences O^n :

$$S^n = \frac{E[(O^n - \mu_{O^n})^3]}{\sigma_{O^n}^3}. \quad (1)$$

A data set having high hubness produces few hub objects with very high n -occurrence and many anti-hubs with n -occurrence of zero. This makes the distribution of n -occurrences skewed with positive skewness indicating high hubness. Previous results [22] show that values above 1.4 are problematic.

Nearest neighbor overlap (L^k): To quantify the degree to which neighborhood relations are preserved we compute the overlap between nearest neighbor lists in the high dimensional input space ($NN(x)$) and the low-dimensional output space ($NN(\hat{x})$):

$$L^k = \frac{1}{N} \sum_{i=1 \dots N} |NN(x) \cap NN(\hat{x})|/k. \quad (2)$$

¹ http://ismir2004.ismir.net/genre_contest/index.htm

Nearest neighbor classification accuracy (C^k): We report the k -nearest neighbor (kNN) classification accuracy C^k using leave-one-out cross-validation, where classification is performed via a majority vote among the k nearest neighbors, with the class of the nearest neighbor used for breaking ties.

4.2 Dimensionality reduction

Dimensionality reduction algorithms try to map high dimensional input data to lower dimensional output spaces while preserving information of the topology of the input space, i.e. preserving similarities or similarity orderings. All three methods used in this study are based on optimization algorithms that are initiated randomly and therefore can give different solutions for different initializations. All results reported in Section 5 are based on single runs since repeated runs have shown that all three methods give comparable solutions even for different initializations. Please note that the original and re-scaled distance matrices D_I and D_G are normalized to have a smallest value of 0 and a largest value of 1 and, if necessary, changed to similarities before dimensionality reduction.

t-Stochastic Neighbor Embedding (TSNE): A particularly successful algorithm for dimensionality reduction is t-SNE [28]. It first converts similarities of high dimensional points x_i and x_j into conditional probabilities $p_{j|i}$ that x_i and x_j are neighbors given a Gaussian probability density estimate centered at x_i . It computes a similar probability $q_{j|i}$ for the low dimensional counterparts y_i and y_j based on a Student-t density estimate. The mapping to the lower dimension is then achieved by minimizing the sum of the Kullback-Leibler divergences over all data points using gradient descent:

$$C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} \quad (3)$$

We used the implementation by Laurens van der Maaten² that accepts similarity matrices as input (function “tsne.p”) using standard settings as provided by the software and 1000 iterations for all experiments.

Sammon mapping (SAMMON): Sammon mapping [17] does dimensionality reduction by minimizing the following via steepest descent:

$$\frac{1}{\sum_{i=0}^{N-1} \sum_{j<i} d(x_i, x_j)} \sum_{i=0}^{N-1} \sum_{j<i} \frac{(d(x_i, x_j) - \hat{d}(\hat{x}_i, \hat{x}_j))^2}{d(x_i, x_j)} \quad (4)$$

where $\hat{d}(\hat{x}_i, \hat{x}_j)$ is the distance in the output space that corresponds to the distance $d(x_i, x_j)$ in the input space and N is the number of points to be mapped. We used the implementation from the SOM Toolbox³ for all experiments with standard settings and 100 iterations.

Self Organizing Map (SOM): The SOM [10] is an unsupervised neural network that visualizes high dimensional

² <http://lvdmaaten.github.io/tsne/>

³ <http://www.cis.hut.fi/projects/somtoolbox/>

data by mapping it to a two dimensional map grid. Data points that are similar in the original high dimensional space are mapped onto locations close to each other on the grid. In essence the SOM consists of an ordered set of so called map units r_i , each of which is assigned a reference vector (or model vector) m_i in the high dimensional input space. In an iterative learning procedure the model vectors m_i are adapted to the input data, very much like cluster centers in a k-means clustering procedure. The main difference is that model vectors corresponding to neighboring map units r_i are adapted together, based on a neighborhood weighting function. This yields a topological organization of the model vectors m_i in the two dimensional output space.

For all our experiments we use SOMs with 40×40 output maps, thereby ensuring that we always have more model vectors than input vectors which is advantageous for using SOM for visualization (see [2] for more on the usage of SOMs for clustering and visualization). We use the NETLAB [12] SOM implementation with standard settings for the learning parameters (initial neighborhood size of 8 shrunk to 1 during an ordering phase lasting 50 iterations, followed by 400 iterations of a convergence phase). Since SOMs need data vectors and not distance matrices as input data, we use the full rows of the distance matrices as inputs (see e.g. [9, 13] for more detail or [22] for a criticism of this rather crude but standard approach).

4.3 Reducing hubness

We introduce the two methods we will apply to reduce hubness by using each method on the whole distance matrix and computing re-scaled distances. Both methods aim at repairing asymmetric nearest neighbor relations which are a consequence of the presence of hubs. A hub y is the nearest neighbor of x , but the nearest neighbor of the hub y is another point a ($a \neq x$). This is because hubs are by definition nearest neighbors to very many data points but only one data point can be the nearest neighbor to a hub.

Mutual Proximity (MP): MP reinterprets the original distance space so that two objects sharing similar nearest neighbors are more closely tied to each other, while two objects with dissimilar neighborhoods are repelled from each other. This is done by transforming the distance of two objects into a mutual proximity in terms of their distribution of distances. It was shown that by using this mutual reinterpretation of distances hubness is decisively reduced, while the intrinsic dimensionality of the data stays the same [20]. To compute MP, we assume that the distances $D_{x,i=1..N}$ from an object x to all other objects in our data set follow a certain probability distribution, thus any distance $D_{x,y}$ can be reinterpreted as the probability of y being the nearest neighbor of x , given their distance $D_{x,y}$ and the probability distribution $P(X)$. In this work we assume that the distances $D_{x,i=1..N}$ follow a Gaussian distribution. MP is defined as the probability that y is the nearest neighbor of x given $P(X)$ and x is the nearest neighbor of y given $P(Y)$:

$$MP(D_{x,y}) = P(X > D_{x,y} \cap Y > D_{y,x}). \quad (5)$$

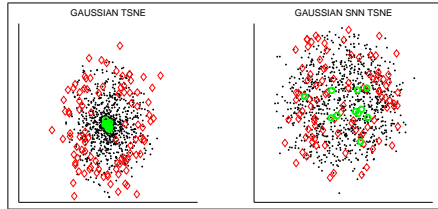


Figure 1. Maps obtained for Gaussian artificial data via TSNE (left) and SNN TSNE (right). Hubs are shown as green circles and anti-hubs as red diamonds.

Shared Nearest Neighbors (SNN): SNN [7] uses the neighbor information to help enforce pairwise stability. SNN is computed as a set intersection of the k -nearest neighbor lists NN of two objects x, y :

$$SNN(x, y) = |NN(x) \cap NN(y)|/k. \quad (6)$$

This way SNN strictly strengthens symmetric nearest neighbor relations which in turn should also manifest itself in a reduction of hubness. We use SNN with $k = 50$ because this already yields hubness values S^5 (see Section 4.1) below 1 for both ISMIR and GTZAN.

5. RESULTS

Before we present our results using the ISMIR and GTZAN data sets we give a first illustration based on artificial data. We sampled 1000 data points from a 50-dimensional Gaussian distribution and used Euclidean distance to compute a distance matrix. The hubness S^5 of this data set is 2.95. Similar to other work [20], we defined anti-hubs as points with a $O^5 = 0$, i.e. points never appearing in any nearest neighbor list of size 5. Hubs are points with $O^5 > 25$, i.e. points that appear more than five times as expected. This definition of hubs and anti-hubs is used for all results in this paper and hubs and anti-hubs are always computed in the high-dimensional spaces. Figure 1 plots two dimensional results obtained using TSNE alone (left plot) and SNN plus TSNE (right plot). As can be seen, TSNE maps all hubs (green circles) to the center of the points and maps all anti-hubs (red diamonds) to the edges. The right plot shows that SNN TSNE is able to map hubs and anti-hubs much more evenly across the whole set of mapped points. The hubness S^5 of the re-scaled distance space after application of SNN is 0.81.

Next we present the visualization results obtained for the ISMIR data set using different combinations of TSNE, SOM, SAMMON and MP and SNN in Figure 2. The hubness S^5 of the ISMIR data set is 3.94. Re-scaling reduces this value to 1.25 for MP and 0.89 for SNN. Hubs are again shown as green circles and anti-hubs as red diamonds. When using TSNE (top row), we again see that the hub points are mapped to the center of the visualization and anti-hubs appearing all over the plot but also at the edges

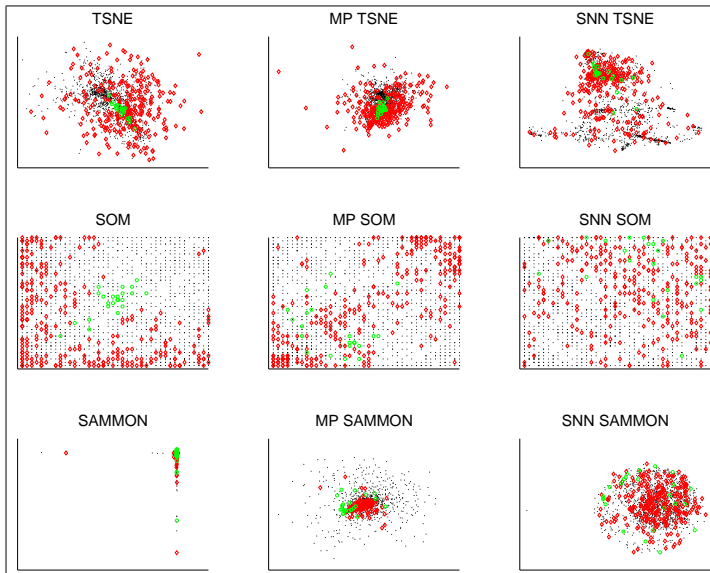


Figure 2. Visualization of ISMIR data set using different combinations of TSNE, SOM, SAMMON and MP and SNN. Hubs are shown as green circles and anti-hubs as red diamonds.

where no other data points are mapped to. When using the combination MP TSNE, this situation shows only little improvement with some anti-hubs still being mapped to areas where no other points can be found. Hub points are still mapped to the center of the plot. When using the combination SNN TSNE the result seems to be much improved, the plot showing much more structure and the hubs and anti-hubs no longer confined to the center or edges. Looking at the results obtained for SOM (middle row), we can again see that the hub points are mapped to the center of the plot whereas the anti-hubs are confined to the left and bottom edge areas. When using MP SOM or even better SNN SOM, hubs and anti-hubs are much more scattered across the whole plots. When using SAMMON (bottom row), we can see that the visualization is heavily distorted with a few data points being mapped far away from the rest of the data. When using MP SAMMON, this distortion is no longer visible but both hubs and anti-hubs are mapped to the more central parts of the plots. Only SNN SAMMON seems to be able to map anti-hubs more or less evenly across the plot, with hubs being mapped closer to the edges. Overall the combination SNN TSNE seems to yield the best visualization results. Results are similar for GTZAN, but are not depicted for lack of space.

To quantify the success in visualization, we compute the nearest neighbor overlap L^k between high- and low-dimensional spaces for TSNE, SOM and SAMMON which is shown in Figure 3 for both ISMIR (top row) and GTZAN

(bottom row) data sets. In all six plots solid lines show results when using dimensionality reduction only (TSNE, SOM or SAMMON), dash-dotted lines give results when MP is used for preprocessing, dashed lines when SNN is used for preprocessing. The overlap L^k is computed for a range of $k = 5 \dots 500$ plotted on the x-axis to quantify preservation of local and more global neighborhoods. We can see that for all three dimensionality reduction methods and over the full range of k , preprocessing via MP and SNN is able to increase the overlap L^k . The only exception is SAMMON when applied to ISMIR, where SNN gives worse results than using no preprocessing for $k > 200$. Preprocessing with SNN is superior to using MP in combination with TSNE and SOM. In combination with SAMMON, MP works a little better than SNN. Overall TSNE performs better than SOM, which is again better than SAMMON. Again the combination SNN TSNE gives the best results of all.

Next we present a more detailed analysis of the nearest neighbor overlap results by concentrating on L^k with $k = 50$ since this is where the difference in performance is largest. In Figure 4 we give separate results for “all” data points, “hub”, “anti-hub” and “normal” (i.e. not hubs or anti-hubs) data points as bar plots for TSNE, SOM and SAMMON. Every bar plot shows a black bar for dimensionality reduction only, a gray bar for results when MP is used for preprocessing, a white bar when SNN is used. For all three dimensionality reduction algorithms, L^{50} is high-

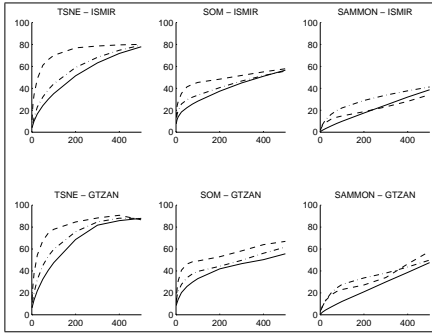


Figure 3. Overlap of nearest neighbors in high and low dimensions in percent (y-axis) vs. number of neighbors (x-axis) for ISMIR (top) and GTZAN (bottom). Plots given for TSNE, SOM and SAMMON with solid lines for using dimensionality reduction only, dash-dotted lines for using MP for preprocessing, dashed lines when SNN is used.

	ISMIR			GTZAN		
	-	mp	snn	-	mp	snn
orig	71.4	78.1	76.5	61.7	67.8	61.6
tsne	62.6	63.9	70.4	39.1	41.0	52.8
som	65.2	65.0	69.2	40.4	37.7	49.7
sammon	47.0	51.3	46.3	16.1	27.3	25.3

Table 1. Genre classification accuracy in percent using 50-nearest neighbor classification for ISMIR and GTZAN data sets different combinations of TSNE, SOM, SAMMON and MP and SNN as well as for the original (orig) high dimensional data space.

est for hub points and worst for anti-hub points, with normal points somewhere in between. Applying either MP or SNN as preprocessing generally increases L^{50} for all different kinds of points, but also makes L^{50} for hubs, anti-hubs and normal points perform much more comparable. Anti-hub points now perform almost as well as all other points. The only exception is again SAMMON, which generally performs very poorly and where SNN is not able to improve the overall situation.

As a further analysis of our visualization results, we give kNN genre classification⁴ accuracy results C^{50} when using different combinations of TSNE, SOM, SAMMON and MP and SNN as well as for the original high dimensional data space in Table 1. As for the original input space (row “orig”), MP and SNN increase C^{50} for ISMIR, but only MP for GTZAN. This is in line with previous comparison of MP and SNN [4]. Classification results for low-dimensional spaces are of course lower than those achieved on the original input spaces since any dimensionality reduction incurs some loss of information. But for

⁴Please note that we are of course aware of the controversial role of genre classification in MIR, especially in the context of GTZAN [25], but that accuracy only serves as a further illustration of results in this context.

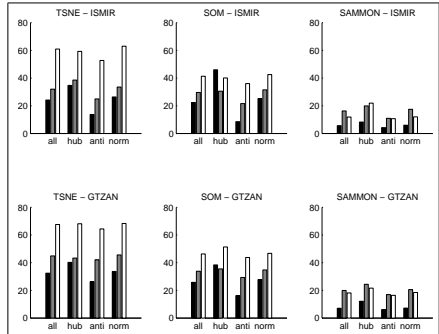


Figure 4. Analysis of overlap of 50 nearest neighbors in high and low dimensions in percent (y-axis) vs. type of data points (x-axis: all, hub, anti-hub, normal) for ISMIR (top) and GTZAN (bottom). Bar plots are given for TSNE, SOM and SAMMON with black bars showing results for dimensionality reduction only, gray bars for using MP for preprocessing, white bars when SNN is used.

both TSNE and SOM on both data sets, SNN is able to increase C^{50} , which is additional indication that SNN is the preprocessing method to prefer. Results for SAMMON are generally very low and rather mixed.

Finally we show visualization results of the ISMIR data set when using TSNE as well as SNN TSNE, which is the best performing combination, in Figure 5 with different genres given in different colors. The color coding is as follows: classical - black, jazz_blues - blue, rock_pop - red, world - green, metal_punk - yellow, electronic - cyan. Although it is hard to verbalize the information contained in a visualization, it seems apparent that the result for SNN TSNE (right plot) shows much more structure than the result for TSNE only. This enables a more detailed picture of the overlap between “classical” (black) and “world” (green) music. Also the position of genre “jazz_blues” (blue) is now clearer between “classical/world” and the remaining three genres. Also “electronic” (cyan) music seems to be a little more apart from “rock_pop” (red) and “metal_punk” (yellow). Results for GTZAN, which consists of music from ten genres, are similar in tendency but not shown for space considerations.

6. DISCUSSION

Summing up the results presented in the previous section, we like to state that all three visualization methods are affected by the hubness problem. Looking at the visualizations, checking the amount of overlap between nearest neighbors in high and low dimensions for hub, anti-hub and normal points makes it clear that there is a problem for dimensionality reduction of data with high values of hubness. It is also evident that preprocessing with either MP or SNN can help in this situation. Especially the combination

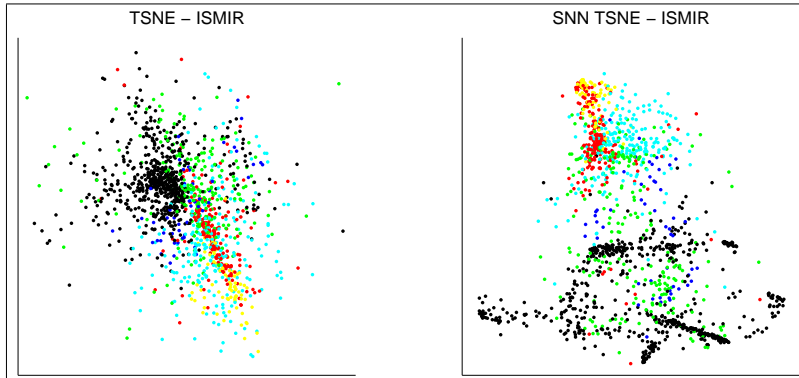


Figure 5. Visualization of ISMIR data set using TSNE (left) and SNN TSNE (right) with color coded genes (see Section 5).

of SNN and TSNE yields very improved results. Although this paper used only one particular approach to compute music similarity, previous work [3] has made it clear that many different approaches are affected by hubness. For the dimensionality reduction algorithms, we basically used standard settings since attempts to adjust parameters did not really improve results. But of course a more rigorous parameter tuning should be part of future research.

One particularity of the music similarity spaces used in this work is the fact they are based on Gaussian models of timbre information and therefore only distances between models are available but not vector representations. Therefore all dimensionality reduction methods need to be able to deal with distance/similarity information as input. Whereas this is natural for SAMMON, it already constitutes a problem for SOM. We resorted to the standard but somewhat crude approach to use the full rows of the distance matrices as input vectors (with length equal 1000 or 1485 for GTZAN and ISMIR). But there already exists a superior approach [22] of directly using Gaussian models as inputs to SOMs and it would be very interesting to research the impact of hubness on this version of SOM. For TSNE, we were able to use a variant (“tsne_p”) that is able to deal directly with similarity matrices. But as stated by the authors [28], this should only be done if “these similarities can be interpreted as conditional probabilities” as explained in Section 4.2. A theoretic examination as to what extent MP and SNN fulfill this requirement will be part of future work. When TSNE is being used with input vectors instead of a similarity matrix, the width of the Gaussian probability densities are adapted locally according to a so-called perplexity term. This is an important part of the algorithm which is missing in case it is used with a similarity matrix directly. It is an interesting research question whether this local adaption in itself is able to counter some of the problems due to hubness. But this can only be studied if vectors are available as input to TSNE.

As has already been noted in Section 3, the music simi-

larity spaces are based on symmetric Kullback-Leibler divergences which do not fulfill the triangle inequality and therefore do not exhibit all aspects of a true metric. There exists an extension of t-SNE [29] which uses multiple maps to visualize non-metric similarities. Even more interesting, this extension is motivated with the notion of data points which show high centrality, i.e. points which are similar to very many other data points. In contrast to the discussion of hub points, such central points are in this case not seen as problematic but as a special challenge for a visualization algorithm. It would therefore be very interesting to study and compare these central and hub points and to apply the t-SNE algorithm for non-metric similarities to data sets with high hubness.

7. CONCLUSION

We presented the first substantial empirical evaluation of the impact of hubness on visualization of high-dimensional music similarity spaces. Analyzing three popular methods for dimensionality reduction applied to two standard music data sets, we were able to show that hubs and anti-hubs distort the lower dimensional representations. Generally hubs are mapped to the central parts of plots and anti-hubs usually to the edges. We were able to show that preprocessing with methods that have been designed to reduce hubness can greatly improve this situation. This results in visualization where hubs and anti-hubs are no longer mapped to peculiar locations, which also gives improved preservation of neighborhood information when mapping to low dimensions. Particularly a combination of preprocessing via “shared nearest neighbors” followed by dimensionality reduction via “t-SNE” proved to be most successful. This approach could therefore be used as the core technology for future visualization interfaces to music catalogs.

Acknowledgements: This work was supported by the Austrian Science Fund (FWF, grant P27082).

8. REFERENCES

- [1] Aucouturier, J.-J., Pachet F.: Improving Timbre Similarity: How high is the sky?, *Journal of Negative Results in Speech and Audio Sciences*, 1(1), 2004.
- [2] Flexer A.: On the Use of Self-organizing Maps for Clustering and Visualization, *Intelligent Data Analysis*, Vol. 5, Number 5, pp. 373-384, 2001.
- [3] Flexer A., Schnitzer D., Schlüter J.: A MIREX meta-analysis of hubness in audio music similarity, *Proceedings of the 13th International Society for Music Information Retrieval Conference*, 2012.
- [4] Flexer A., Schnitzer D.: Can Shared Nearest Neighbors Reduce Hubness in High-Dimensional Spaces?, *Proceedings of 1st International Workshop on High Dimensional Data Mining (HDM)*, IEEE International Conference on Data Mining, 2013.
- [5] Gasser M., Flexer A.: FM4 Soundpark: Audio-based Music Recommendation in Everyday Use, *Proc. of the 6th Sound and Music Computing Conference*, 2009.
- [6] Gasser M., Flexer A., Schnitzer D.: Hubs and Orphans - an Explorative Approach, *Proceedings of the 7th Sound and Music Computing Conference*, 2010.
- [7] Jarvis R., Patrick E.A.: Clustering using a similarity measure based on shared near neighbors, *IEEE Transactions on Computers*, vol. 22, pp. 1025-1034, 1973.
- [8] Karydis I., Radovanović M., Nanopoulos A., Ivanović M.: Looking through the "glass ceiling": A conceptual framework for the problems of spectral similarity, *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pp. 267-272, 2010.
- [9] Knees P., Schedl M., Pohle T., Widmer G.: Exploring music collections in virtual landscapes, *IEEE multimedia*, 14(3):4654, 2007.
- [10] Kohonen T.: *Self-Organizing Maps*, Springer, 2001.
- [11] Mandel M., Ellis D.: Song-level features and support vector machines for music classification, *Proc. of the 6th Int. Conf. on Music Information Retrieval*, 2005.
- [12] Nabney I.: *NETLAB: Algorithms for Pattern Recognition*, Springer Science & Business Media, 2002.
- [13] Pampalk E.: *Computational models of music similarity and their application in music information retrieval*, Doctoral dissertation, Vienna University of Technology, Austria, 2006.
- [14] Pampalk E., Dixon S., Widmer G.: Exploring music collections by browsing different views, *Computer Music Journal*, Vol. 28, No. 2, pp. 49-62, 2004.
- [15] Radovanović M., Nanopoulos A., Ivanović M.: Hubs in space: Popular nearest neighbors in high-dimensional data, *Journal of Machine Learning Research*, 11:2487-2531, 2010.
- [16] Rauber A., Frühwirth M.: Automatically Analyzing and Organizing Music Archives, *Proceedings of the European Conference on Research and Advanced Technology for Digital Libraries*, pp. 4-8, 2001.
- [17] Sammon J.W.: A nonlinear mapping for data structure analysis, *IEEE Transactions on Computers*, vol. C-18, no. 5, pp. 401-409, 1969.
- [18] Schedl M., Flexer A., Urbano J.: The Neglected User in Music Information Retrieval Research, *Journal of Intelligent Information Systems*, 41(3), 523-539, 2013.
- [19] Schnitzer D., Flexer A.: The Unbalancing Effect of Hubs on K-medoids Clustering in High-Dimensional Spaces, *Proceedings of the International Joint Conference on Neural Networks (IJCNN)*, 2015.
- [20] Schnitzer D., Flexer A., Schedl M., Widmer G.: Local and Global Scaling Reduce Hubs in Space, *Journal of Machine Learning Research*, 13:2871-2902, 2012.
- [21] Schnitzer D., Flexer A., Tomašev N.: A Case for Hubness Removal in High-Dimensional Multimedia Retrieval, *Proceedings of the 36th European Conference on Information Retrieval (ECIR)*, 2014.
- [22] Schnitzer D., Flexer A., Widmer G., Gasser M.: Islands of Gaussians: The Self Organizing Map and Gaussian Music Similarity Features, *Proceedings of the Eleventh International Society for Music Information Retrieval Conference (ISMIR'10)*, 2010.
- [23] Serra X., Magas M., Benetos E., Chudy M., Dixon S., Flexer A., Gomez E., Gouyon F., Herrera P., Jorda S., Paytuvi O., Peeters G., Schlüter J., Vinet H., Widmer G., *Roadmap for Music Information Research*, Peeters G. (editor), 2013.
- [24] Stober S., Nürnberger A.: MusicGalaxy - an adaptive user-interface for exploratory music retrieval, *Proceedings of the 11th International Conference on Music Information Retrieval*, 2010.
- [25] Sturm B. L.: Classification accuracy is not enough, *Journal of Intelligent Information Systems*, 41(3), 371-406, 2103.
- [26] Tomašev N., Radovanović M., Mladenčić D., Ivanović M.: The Role of Hubness in Clustering High-dimensional Data, *IEEE Transactions on Knowledge and Data Engineering*, Volume 26, Issue 3, 2013.
- [27] Tzanetakis G., Cook P.: Musical genre classification of audio signals, *IEEE Transactions on Speech and Audio Processing*, Vol. 10, Issue 5, 293-302, 2002.
- [28] van der Maaten L.J.P., Hinton G.E.: Visualizing High-Dimensional Data Using t-SNE, *Journal of Machine Learning Research*, 9(Nov):2579-2605, 2008.
- [29] van der Maaten L.J.P., Hinton G.E.: Visualizing Non-Metric Similarities in Multiple Maps, *Machine Learning*, 87(1):33-55, 2012.

I-VECTORS FOR TIMBRE-BASED MUSIC SIMILARITY AND MUSIC ARTIST CLASSIFICATION

Hamid Eghbal-zadeh Bernhard Lehner Markus Schedl Gerhard Widmer
Department of Computational Perception, Johannes Kepler University of Linz, Austria

hamid.eghbal-zadeh@jku.at

ABSTRACT

In this paper, we present a novel approach to extract song-level descriptors built from frame-level timbral features such as Mel-frequency cepstral coefficient (MFCC). These descriptors are called identity vectors or *i-vectors* and are the results of a factor analysis procedure applied on frame-level features. The *i-vectors* provide a low-dimensional and fixed-length representation for each song and can be used in a supervised and unsupervised manner.

First, we use the *i-vectors* for an unsupervised music similarity estimation, where we calculate the distance between *i-vectors* in order to predict the genre of songs.

Second, for a supervised artist classification task we report the performance measures using multiple classifiers trained on the *i-vectors*.

Standard datasets for each task are used to evaluate our method and the results are compared with the state of the art. By only using timbral information, we already achieved the state of the art performance in music similarity (which uses extra information such as rhythm). In artist classification using timbre descriptors, our method outperformed the state of the art.

1. INTRODUCTION AND RELATED WORK

In content-based music similarity and classification, acoustic features are extracted from audio and characteristics of a song are projected into a new space called feature space. In this space, different attributes can be captured based on the features used. For example, features such as Fluctuation Pattern (FP) [26], reflect the variability related to the rhythm; and features such as MFCCs, demonstrate the timbral perspective of a song. However, the diversity of music genres, the presence of different musical instruments and singing techniques make the capturing of these variabilities difficult. Different modeling techniques and machine learning approaches are used to find the factors in the feature space that best represent these variabilities.

Multiple approaches have been followed in the literature for extracting the features from songs in which 1) clas-

sical frame-level features, 2) block-level features and 3) song-level features are the most frequently used methods in MIR.

1.1 Frame-level features

In the frame-level approach, features are often extracted from short-time frames of a song. In this approach, frames are first classified directly, and then the results are combined to make a decision for a song.

1.2 Block-level features


Block-level features process the frames in terms of blocks, where each block consists of a fixed number of frames. They are built in two steps: first, the block processing step and second, the generalization step. In the first step, by selecting a collection of frames using a pattern, blocks are built. Then in the second step, the feature values of all blocks are combined into a single representation for the whole song. In [29], six different block-level features are introduced and a method is proposed to fuse all the blocks together. Block-level features [5, 24, 26, 29] have shown considerable performances in the MIREX¹ challenges.

1.3 Song-level features

Song-level features are found useful in artist recognition as well as music similarity estimation. In [30], a compact signature is generated for each song, and then is compared to the other songs using a graph matching approach for artist recognition. In [21] multivariate kernels have been used to model an artist. Recently, [5, 29] proposed methods to extract a fixed-length vector from a song to be used in music similarity estimation and genre classification.

The advantage of methods based on song-level features is that different tools such as dimensionality reduction (e.g. Principal Components Analysis (PCA) [15]) and projections can be applied to songs. For example, in [5], super-vectors extracted via a Gaussian Mixture Model (GMM) are found useful to represent songs and calculate the similarity using Euclidean distance. In [24] a method using song-level features is presented, which models frame-level descriptors such as MFCCs and FP with a single Gaussian and then the similarity between songs is calculated using Kullback Leibler divergence. In [26], rhythm descriptors

¹ Annual Music Information Retrieval eXchange (MIREX). More information is available at: <http://www.music-ir.org>

 © Hamid Eghbal-zadeh, Bernhard Lehner, Markus Schedl, Gerhard Widmer. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Hamid Eghbal-zadeh, Bernhard Lehner, Markus Schedl, Gerhard Widmer. "I-Vectors for Timbre-Based Music Similarity and Music Artist Classification", 16th International Society for Music Information Retrieval Conference, 2015.

are introduced to improve the performance of music similarity measures in [24].

1.3.1 GMM and GMM-supervectors

GMMs have been frequently used for acoustic modeling in music processing [4, 5, 12]. In [4, 5], a GMM is used as a *Universal Background Model* (UBM) for content-based music similarity estimation and genre classification.

Gaussian-based features used in [5, 24] are other examples of song-level features which use a Gaussian model to create a statistical representation of a song from frame-level features. Similar to [4, 5], a GMM supervector is computed for each song. This representation is a fixed-length vector, and is computed using a UBM (which is a GMM, trained on a database of songs) via a procedure described in [4, 5].

The first drawback of GMM-based methods is that when the rank of the GMM space (number of Gaussian components) increases, the dimensionality of GMM supervectors rises which causes problems such as the curse of dimensionality. One solution to this issue would be to use dimensionality reduction methods such as PCA. In our previous work [9], we showed that this is not effective. Another solution would be to decompose these high-dimensional supervectors into multiple terms with lower ranks which we will discuss in the following section.

1.3.2 Session and Speaker variability

As described in [18], there exists a second drawback of GMM-based methods. The performance of these frameworks suffer from their inability to capture the variability known as *session variability* in the field of speaker verification. In contrast to *speaker variability* which is the variability that appears between different speakers, *session variability* is defined as the variability that appears for a speaker from one recording to another [18]. This variability is called *session* because it appears inside a recording session of a speaker.

1.3.3 Song, Genre and Artist variability

In MIR, similar to session variability, we define *song variability* as the variability that appears between songs. Also, similar to speaker variability, we define *genre variability* for genre classification as the variability that appears between different genres, and *artist variability* for artist recognition as the variability appears between different artists.

The second drawback of GMM-based methods is that they can not distinguish between song variability and genre (or artist) variability. If we can provide a decomposition of GMM supervectors in a way that separates the desired factors, such as genre variability from undesired ones, such as song variability, and at the same time decreases the dimensionality of GMM supervectors, then as a result a better representation of GMM supervectors with lower dimensionality and better discrimination power will be obtained. Factor Analysis (FA) provides the means to produce such representations where a GMM supervector de-

composes into multiple factors. An advantage of the features obtained by FA compared to block-level features and Gaussian-based features is that FA can be performed in a way that after decomposition, each component can exhibit a specific variability such as artist or genre. Thus, desired factors can be kept and undesired factors can be removed from the song's GMM supervector. By applying such decomposition on top of the GMM space, another space with bases of desired factors (e.g. genre space, with genre factors) can be created.

Recently, in the field of speaker verification, Dehak et al. [7] introduced **i-vectors** which outperformed the state of the art and provided a solution for the problem of session variability in the GMM-UBM frameworks. The i-vector extraction is a feature-modeling technique that builds utterance-level features, and it has been successfully used in other areas such as emotion recognition [34], language recognition [8], accent recognition [1] and audio scene detection [10].

The i-vector method applies a FA procedure to extract low-dimensional features from GMM supervectors. This FA procedure estimates hidden variables in GMM supervector space, which provides better discrimination ability and lower dimensionality than GMM supervectors. These hidden variables are the i-vectors and even though the **i-vector extraction procedure is totally unsupervised**, they can be used for both supervised and unsupervised tasks. The aim of this paper is to introduce the i-vectors to the MIR community and show their performance on two of the major tasks in content-based MIR.

2. FACTOR ANALYSIS PROCEDURE

In this paper, examples are given from a **genre classification** point of view. The definitions and the method are extendable to other tasks in MIR such as artist classification.

2.1 Overview of Factor Analysis Methods

A FA model can be viewed as a GMM supervector space, where genre and song factors are its hidden variables. Genre and song factors are defined in a way that for a given genre, the values of the genre factors are assumed to be identical for all songs within that genre. The song factors may vary from one song to another.

Let's assume we have a C mixture components GMM and let F be the dimension of the acoustic feature vectors. For each mixture component $c = 1, \dots, C$, let m_c denote the corresponding genre-independent mean vector (UBM mean vector) and let m denote the $C \cdot F \times 1$ supervector obtained by concatenating m_1, \dots, m_C .

Maximum a posteriori (MAP) [14] is a method that is used to extract genre-dependent GMM supervectors. In MAP, it is assumed that each genre g can be modeled only by a single genre-dependent GMM supervector $M(g)$. This supervector is calculated from a genre-independent vector m which is then adapted to a couple of songs from a specific genre known as the genre-adaptation data.

Similar to speaker modeling in speaker verification [19], the MAP approach to genre modeling assumes that for each mixture component c and genre g , there is an unobservable offset vector O_g such that:

$$M(g) = m + O_g \quad (1)$$

O_g is unknown and can not be learned during the MAP training procedure.

Further, *eigenvoice MAP* [17] assumes the row vectors of the matrix O_g are independent and identically distributed. A rectangular matrix V of dimensions $C \cdot F \times R$ is assumed where R is a parameter such that $R \ll C \cdot F$. The V matrix has a lower rank than $C \cdot F$ and can be learned from the training data. The supervector $M(g)$ decomposes into factors $y(g)$ which have lower ranks using V . For genre g , the FA used in eigenvoice MAP is as follows:

$$M(g) = m + Vy(g) \quad (2)$$

where $y(g)$ is a hidden $R \times 1$ vector which has a standard normal distribution. Eigenvoice MAP trains faster than MAP, yet training V properly needs a very large amount of data, also song factors are not considered in the decomposition of $M(g)$.

A solution for separation between song and genre factors was first suggested in [19], and later improved in [16] as Joint Factor Analysis (JFA). JFA decomposition model can be written as follows:

$$M = m + Vy + Ux + Dz \quad (3)$$

where M is a song GMM supervector, m is a genre- and song-independent supervector which can be calculated using a UBM, V and D define a genre subspace (genre matrix and diagonal residual, respectively), and U defines a song subspace. The vectors y, z are the genre-dependent factors, and x is the song-dependent factor in their respective subspaces. They are assumed to be a random variable with a standard normal distribution. Unlike eigenvoice MAP, JFA gives us a modeling with separated genre and song factors with low ranks, where they can be used to better separate songs from different genres by removing song variability.

Even though JFA showed better performance than previous FA methods, in terms of separation between song and genre factors, experimental results in [6] proved that if we extract song and genre factors using JFA, song factors also contain information about genres. Based on this finding, another FA model is proposed in [7], which defines a new low-dimensional space called Total Variability Space (TVS). The vectors in this new space, are called i-vectors. In the TVS, both song and genre factors are considered, but modeled together as a new factor named *total factor*. Total factors have lower dimensionality than GMM supervectors and one can represent a song by extracting total factors from its GMM supervector. Because i-vector FA showed the best results in speaker verification [7], in this paper we use it for multiple tasks in MIR. The FA procedure used to obtain i-vectors is described in the next section.

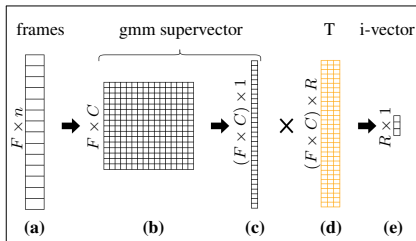


Figure 1: Graphical representation of different vectors extracted during i-vector FA. F is the dimensionality of acoustic features, C is the number of Gaussian components, and R is the rank of the TVS matrix. a) frame-level features of a song. b) and c) GMM supervector. d) TVS matrix T . e) i-vector.

2.2 Overview of I-vectors

TVS refers to total factors that contain both genre and song factors. In the TVS, a given song is represented by a low-dimensional vector called **i-vector**, which provides a good genre separability. This i-vector is known as point estimate of the hidden variables in a FA model similar to JFA. This describes these hidden variables and their characteristics.

In Figure 1, a graphical representation of vectors used in different steps during i-vector FA is provided. From each song, first frame-level features of dimensionality F are extracted as shown in Figure 1-a. Then, a C mixture components GMM trained on a large number of songs is used to extract GMM supervectors of dimension $F \times C$. This rectangular vector (Figure 1-b) then reshapes to a $(F \cdot C) \times 1$ vector (Figure 1-c). A matrix of $(C \cdot F) * R$ known as TVS matrix (T) is learned from a set of songs. T matrix is used to reduce the dimensionality of GMM supervectors to R where R is the rank of T , as can be observed in Figure 1-d. The resulting vectors are i-vectors having a low rank of R (Figure 1-e).

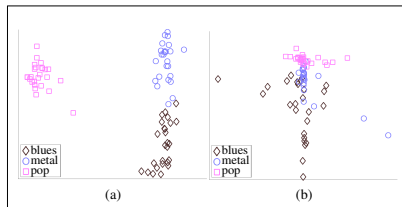


Figure 2: 2D PCA projected vectors extracted from songs of 3 different genres in GTZAN dataset. a) i-vectors. b) GMM supervectors.

A comparison between GMM representation and i-vector representation is provided in Figure 2. This visualization is prepared by projecting GMM supervectors and i-vectors using PCA into a 2 dimensional plane. Multiple

songs of 3 different genres from the GTZAN dataset² are selected, then both their GMM supervectors and i-vectors are extracted. In Figure 2-a, a scatter plot of the song's projected i-vectors are shown. Also, in Figure 2-b, GMM supervectors projected using PCA are displayed. It can be observed that i-vector extraction was successful at increase the discrimination between songs of different genres. In the following paragraphs, the i-vector FA is described.

A C mixture components GMM ($c = 1, \dots, C$) called UBM can be trained on a large amount of data from multiple genres, where for component c , w_c , m_c and Σ_c denote mixture weight, mean vector and covariance matrix respectively. Given a song of genre g , a GMM supervector $M(g)$ can be calculated from a sequence of X_1, \dots, X_T frames. The i-vector FA equation decomposes the vector $M(g)$ as follows:

$$M_c(g) = m_c + Ty \quad (4)$$

where $M_c(g)$ corresponds to a subvector of $M(g)$ for component c , m_c is the genre- and song-independent vector, and $y \sim \mathcal{N}(0, 1)$ is the genre- and song-dependent vector, known as the i-vector. A rectangular matrix T of low rank known as TVS matrix is used to extract i-vectors from the vector $M_c(g)$.

The i-vector y is a hidden variable, but we can find it using the mean of its posterior distribution. This posterior distribution is Gaussian and is conditioned to the BaumWelch (BW) statistics for a given song [17]. The zero-order and the first-order BW statistics used to estimate y , are called N_c and P_c respectively (see Equation 6). Similar to [20], the BW statistics are extracted using the UBM as follows.

A closed form of an i-vector y looks as follows:

$$y = (I + T^t \Sigma^{-1} N(s) T)^{-1} \cdot T^t \Sigma^{-1} P(s) \quad (5)$$

where we define $N(s)$ as a diagonal matrix of dimension $C \cdot F \times C \cdot F$ with $N_c \times I$ ($c = 1, \dots, C$ and I has $F \times F$ dimensions) diagonal blocks. $P(s)$ is a vector with $C \cdot F \times 1$ dimensions and is made by concatenating all first-order BW statistics P_c for a given song s ; also Σ is a diagonal covariance matrix of dimension $C \cdot F \times C \cdot F$ estimated during the factor analysis procedure; it models the residual variability not captured by the TVS matrix T . The BW statistics N_c and P_c are defined as follows.

Suppose we have a sequence of frames X_1, \dots, X_T and a UBM with C mixture components defined in a feature space of dimension F . The BW statistics needed to estimate the i-vector for a given song are obtained by:

$$\begin{aligned} N_c &= \sum_t \gamma_t(c) \\ P_c &= \sum_t \gamma_t(c) X_t \end{aligned} \quad (6)$$

where, for time t , $\gamma_t(c)$ is the posterior probability of X_t generated by the mixture component c of the UBM.

² <http://marsyas.info/downloads/datasets.html>

Since BW statistics are calculated using a GMM, they are called **GMM supervectors** in i-vector modeling.

TVS matrix T is estimated via an expectation maximization procedure using BW statistics. More information about the training procedure of T can be found in [7, 22].

3. I-VECTORS FOR UNSUPERVISED MUSIC SIMILARITY ESTIMATION

In this section, i-vectors are used for music similarity estimation task. Genre and song variability are the factors used in this task.

3.1 Dataset

The 1517Artists³ dataset is used for training UBM and T matrix. This dataset consists of freely available songs and contains 3180 tracks by 1517 different artists distributed over 19 genres. The GTZAN dataset is used for music similarity estimation which contains 1000 song excerpts of 30 seconds, evenly distributed over 10 genres.

3.2 Frame-level Features

We use MFCCs as one of our timbral features. MFCCs are the most utilized timbre-related frame-level features in MIR. They are a compact, and perceptually motivated representation of the spectral envelope.

For the extraction of the MFCCs, we use an observation window of 10 ms, with an overlap of 50%. We extract 25 MFCCs with the rastamat toolbox [11]. The first and second order derivatives (deltas and double-deltas) of the MFCCs are also added to the feature vector.

Additionally, we use the first order derivative of a cent-scaled spectrum, calculated in the same way as explained in [29]. These features are called Spectrum Derivatives (SD).

3.3 Baselines

Four different baselines are used to be compared to our method. The first baseline is fusing block-level similarity measure (BLS) [29], which uses 6 different block-level features containing spectral pattern, delta spectral pattern, variance delta spectral pattern, logarithmic fluctuation pattern, correlation pattern and spectral contrast pattern. These features are used with a similarity function and a distance normalization method to calculate a pairwise distance matrix between songs. The second baseline is called Rhythm Timbre Bag of Features (RTBOF) [26]. RTBOF has two components of rhythm and timbre which are modeled over local spectral features. The third baseline is MARSYAS (Music Analysis, Retrieval and Synthesis for Audio Signals) which has an open source toolbox to calculate various audio features.⁴ A similarity function is used to calculate a distance matrix of features extracted as described in [32]. The last baseline (CMB) is a combination of BLS and RTBOF, which reported in [29] as the best similarity method in case of genre classification measures.

³ This dataset can be downloaded from www.seyerlehner.info.

⁴ <http://marsyas.info>

3.4 Experimental Setup

A UBM with 1024 Gaussian components is trained on the 1517Artists dataset using 2000 consecutive frames from the middle area of each song. No labels are used during the training procedure of UBM and T matrix. The TVS matrix T is trained using 400 total factors, and used during the i-vector extraction procedure. The number of factors and Gaussian components was chosen after a parameter analysis step on a small development dataset which differs from the datasets used in this paper.

Two sets of different i-vectors are used to calculate two similarity matrices for the GTZAN dataset. First, MFCC features are used to extract i-vectors, and cosine distance is used to calculate a pair-wise distance matrix between all songs, since in [7] cosine distance has been successfully used with i-vectors. UBM and T matrix are also trained using MFCC features of 1517Artists.

Second, SD features are used to extract another set of i-vectors to calculate our second distance matrix using cosine distance. Similar to MFCC i-vectors, a new UBM and T matrix is trained using SD features extracted from 1517Artist dataset.

Pair-wise distance matrices are normalized using a distance space normalization (DSN) proposed in [25]. The distance matrices for baseline methods are downloaded from the website⁵ of the author of [29].

3.5 Evaluation

We evaluate the music similarity measures using genre classification via k-nearest neighbor (KNN) classification. This method is also used in [5, 24, 26, 29]. We use different values of k that vary from 1 to 20. Also, we use a leave-one-out scenario for genre classification using pair-wise distance matrices.

3.6 Results and Discussion

The KNN genre classification accuracy calculated using our method is compared to the baseline methods, and the results are shown in Figure 3. As can be seen, our method using MFCC features achieved the performance of the BLS baseline and outperformed MARSYAS. By combining the distance matrices calculated using MFCC and SD i-vectors with equal weights after applying DSN, we could achieve the performance of RTBOF baseline.

Since the authors of the BLS method in [29] reported a combination of BLS and RTBOF (named as CMB in [29]) to perform best, we also combined our MFCC+SD i-vector distance matrix with RTBOF with equal weights after applying DSN and achieved the performance of CMB. Furthermore, by combining MFCC+SD i-vector and CMB distance matrix (with equal weights after DSN), we could achieve a better performance than the best combined method reported in [29].

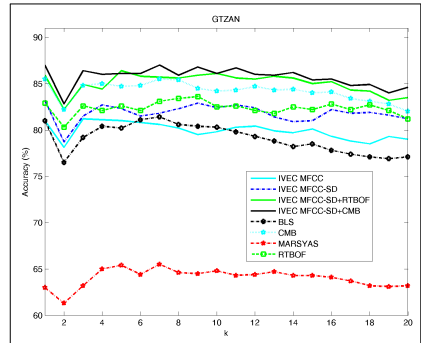


Figure 3: Evaluation results of KNN genre classification on GTZAN dataset.

3.7 Resources

The MSR Identity Toolbox [28] was used for i-vector extraction. We also used drtoolbox [33] to apply PCA for visualization in Figure 2.

4. I-VECTORS FOR SUPERVISED ARTIST CLASSIFICATION

In this section, i-vectors are used for artist recognition task. Artist and song variability are the factors used in this task. More details about artist recognition using i-vectors can be found in our previous work [9].

4.1 Dataset

The artist classification experiments were conducted using the artist20 dataset [12]. It contains 1413 tracks, mostly rock and pop songs, composed of six albums from each of the 20 different artists.

4.2 Frame-level Features

Instead of extracting the MFCCs ourselves, we use the ones provided as part of the dataset in [12]. Neither first nor second order derivatives of the MFCCs are used. Similar to the approach already discussed in Section 3.2, we also include the first order derivative of a cent-scaled spectrum (SD features).

4.3 Baselines

Multiple baseline methods from the literature are selected and their performance is compared to that achieved by our method. Results are reported for a 20-class artist classification task on the artist20 dataset [12]. The first baseline (*BLGMM*) models artists with GMMs using MFCCs [12]. The second baseline (*BLsparse*) uses a sparse feature learning method [31] of ‘bag of features’ (BOF). Both the magnitude and phase parts of the spectrum are used in this

⁵ www.seyerlehner.info

method. The third baseline is (*BLsign*). It generates a compact signature for each song using MFCCs, and then compares these by a graph matching technique [30]. The fourth baseline (*BLmultiv*) uses multivariate kernels [21] with the direct uniform quantization of the MFCC features. The results for the latter three are taken from their publications, while the results for the *BLGMM* baseline are reproduced using the implementation provided with the dataset. The performance of all baselines on the artist20 dataset are reported using the same songs, and the same fold splits in the 6-fold cross-validation.

4.4 Experimental Setup

Similar to the setup followed in Section 3.4, a UBM with 1024 Gaussian components and a T matrix with 400 factors are used for i -vector extraction. Unlike the setup in music similarity estimation, no other dataset is used to train T and the UBM. Instead, in each fold the training set is used to train UBM and T matrix. Unlike the setup described in Section 3.4, we apply a Linear Discriminant Analysis (LDA) [23] to the i -vectors to reduce the dimensionality from 400 to 19. The reason we didn't use LDA for music similarity estimation is that the whole procedure of i -vector extraction in Section 3 was unsupervised, and no labels were used during the i -vector extraction process.

In each fold, the LDA is trained on the same data that UBM and T matrix are trained. i -vectors are centered by removing the mean calculated from training i -vectors, then length-normalized [13] before applying LDA. After applying LDA, once again i -vectors are length-normalized since iterative length-normalization was found useful in [2]. The length normalization provides a standard form of i -vectors.

We fuse MFCC and SD i -vectors of a song simply by concatenating the dimensionality-reduced i -vectors and subsequently feed them into the classifiers investigated.

First, a Probabilistic Linear Discriminant Analysis (PLDA) [27] is used to find the artist for each song (iv-PLDA). PLDA is a generative model which models both intra-class and inter-class variance as multidimensional Gaussian and showed significant results with i -vectors [3]. Second, a KNN classifier with $k = 3$ (3NN) and a cosine distance is considered (iv3NN). Third, a Discriminant Analysis (DA) classifier is investigated with a linear discriminant function and a uniform prior (ivDA).

4.5 Evaluation

A 6-fold cross-validation proposed in [12] is used to evaluate the artist classification task. In each fold, five albums from each artist are used for training and one for testing. We report mean class-specific accuracy, F1, precision and recall, all averaged over folds.

4.6 Results and Discussion

The results of artist classification are reported in Table 1. Using MFCC i -vectors, our proposed method outperformed all the baselines with all three classifiers. Also by

using MFCC+SD i -vectors, the results of artist classification from all 3 classifiers improved. The best artist classification performance is achieved using MFCC+SD i -vectors and a DA classifier yielding 11 percentage point improvement in accuracy and 10 percentage point improvement in F1 compared to the best known results among all the baselines.

Method	Feat.	Acc%	F1%	Pr%	Rec%
BLGMM	20mfcc	55.90	55.18	58.74	58.20
BLsparse	BOF	67.50	n/a	n/a	n/a
BLsign	15mfcc	71.50	n/a	n/a	n/a
BLmultiv	13mfcc	74.30	74.79	n/a	n/a
ivPLDA	20mfcc	83.30	82.58	83.72	84.02
iv3NN	20mfcc	82.43	81.70	83.06	83.03
ivDA	20mfcc	83.36	82.67	84.07	83.78
ivPLDA	20mfcc+sd	85.27	84.58	85.87	85.68
iv3NN	20mfcc+sd	83.68	83.05	84.10	84.55
ivDA	20mfcc+sd	85.45	84.59	85.80	85.68

Table 1: Artist classification results for **different methods** on the **artist20** dataset.

4.7 Resources

We used the same resources as reported in Section 3.7. In addition, we used the PLDA implementation from MSR Identity Toolbox [28] and LDA from drtoolbox [33].

5. CONCLUSION

In this paper, we propose an i -vector based factor analysis (FA) technique to extract song-level features for unsupervised music similarity estimation and supervised artist classification. In music similarity estimation, our method achieved the performance of state-of-the-art methods by using only timbral information. In artist classification, our method was evaluated on a variety of classifiers and proved to yield stable results. The proposed method outperformed all the baselines on the artist20 dataset and improved the best known artist classification measures among baselines. To the best of our knowledge, our results are the highest artist classification results published so far for the artist20 dataset.

6. ACKNOWLEDGMENT

We would like to acknowledge the tremendous help by Dan Ellis from Columbia University, who shared the details of his work, which enabled us to reproduce his experiment results. Thanks to Pavel Kukša from University of Pennsylvania for sharing the details of his work with us. Also thank to Jan Schlüter from OFAI for his help with music similarity baselines. And at the end, we appreciate helpful suggestions of Rainer Kelz and Filip Korzeniowski from Johannes Kepler University of Linz to this work. This work was supported by the EU-FP7 project no.601166 (PHENICX), and by the Austrian Science Fund (FWF) under grants TRP307-N23 and Z159.

7. REFERENCES

- [1] Mohamad Hasan Bahari, Rahim Saiedi, David Van Leeuwen, et al. Accent recognition using i-vector, gaussian mean supervector and gaussian posterior probability supervector for spontaneous telephone speech. In *ICASSP*. IEEE, 2013.
- [2] Pierre-Michel Bousquet, Driss Matrouf, and Jean-François Bonastre. Intersession compensation and scoring methods in the i-vectors space for speaker recognition. In *INTERSPEECH*, 2011.
- [3] Lukas Burget, Oldrich Plchot, Sandro Cumani, Ondrej Glembek, Pavel Matejka, and Niko Brummer. Discriminatively trained probabilistic lda for speaker verification. In *ICASSP*. IEEE, 2011.
- [4] Chuan Cao and Ming Li. Thinkits submissions for mirex2009 audio music classification and similarity tasks. In *MIREX*. Citeseer, 2009.
- [5] Christophe Charbuillet, Damien Tardieu, Geoffroy Peeters, et al. Gmm-supervector for content based music similarity. In *DAFx, Paris, France*, 2011.
- [6] Najim Dehak. *Discriminative and generative approaches for long-and short-term speaker characteristics modeling: application to speaker verification*. Ecole de Technologie Supérieure, 2009.
- [7] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet. Front-end factor analysis for speaker verification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 2011.
- [8] Najim Dehak, Pedro A Torres-Carrasquillo, Douglas A Reynolds, and Reda Dehak. Language recognition via i-vectors and dimensionality reduction. In *INTERSPEECH*. Citeseer, 2011.
- [9] H Eghbal-zadeh, M Schedl, and G Widmer. Timbral modeling for music artist recognition using i-vectors. In *EUSIPCO*, 2015.
- [10] Benjamin Elizalde, Howard Lei, and Gerald Friedland. An i-vector representation of acoustic environments for audio-based video event detection on user generated content. In *ISM*. IEEE, 2013.
- [11] Daniel PW Ellis. PLP and RASTA (and MFCC, and inversion) in Matlab, 2005. online web resource.
- [12] Daniel PW Ellis. Classifying music audio with timbral and chroma features. In *ISMIR*, 2007.
- [13] Daniel Garcia-Romero and Carol Y Espy-Wilson. Analysis of i-vector length normalization in speaker recognition systems. In *INTERSPEECH*, 2011.
- [14] Jean-Luc Gauvain and Chin-Hui Lee. Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains. *Speech and audio processing, IEEE Transactions on*, 1994.
- [15] Ian Jolliffe. *Principal component analysis*. Wiley Online Library, 2002.
- [16] Patrick Kenny. Joint factor analysis of speaker and session variability: Theory and algorithms. *CRIM, Montreal, (Report) CRIM-06/08-13*, 2005.
- [17] Patrick Kenny, Gilles Boulianne, and Pierre Dumouchel. Eigenvoice modeling with sparse training data. *Speech and Audio Processing, IEEE Transactions on*, 2005.
- [18] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel. Speaker and session variability in gmm-based speaker verification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 2007.
- [19] Patrick Kenny, Mohamed Mihoubi, and Pierre Dumouchel. New map estimators for speaker recognition. In *INTERSPEECH*, 2003.
- [20] Patrick Kenny, Pierre Ouellet, Najim Dehak, Vishwa Gupta, and Pierre Dumouchel. A study of interspeaker variability in speaker verification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 2008.
- [21] Pavel P. Kuksa. Efficient multivariate kernels for sequence classification. *CoRR*, 2014.
- [22] Driss Matrouf, Nicolas Scheffer, Benoit GB Fauve, and Jean-François Bonastre. A straightforward and efficient implementation of the factor analysis model for speaker verification. In *INTERSPEECH*, 2007.
- [23] Sebastian Mika, Gunnar Ratsch, Jason Weston, Bernhard Scholkopf, and Klaus-Robert Muller. Fisher discriminant analysis with kernels. In *Signal Processing Society Workshop Neural Networks for Signal Processing*, 1999.
- [24] Elias Pampalk. Audio-based music similarity and retrieval: Combining a spectral similarity model with information extracted from fluctuation patterns. In *ISMIR*, 2006.
- [25] Tim Pohle and Dominik Schnitzer. Striving for an improved audio similarity measure. *Music information retrieval evaluation exchange*, 2007.
- [26] Tim Pohle, Dominik Schnitzer, Markus Schedl, Peter Knees, and Gerhard Widmer. On rhythm and general music similarity. In *ISMIR*, 2009.
- [27] Simon JD Prince and James H Elder. Probabilistic linear discriminant analysis for inferences about identity. In *Computer Vision, ICCV*. IEEE, 2007.
- [28] Seyed Omid Sadjadi, Malcolm Slaney, and Larry Heck. Msr identity toolbox-a matlab toolbox for speaker recognition research. *Microsoft CSRC*, 2013.
- [29] Klaus Seyerlehner, Gerhard Widmer, and Tim Pohle. Fusing block-level features for music similarity estimation. In *DAFx*, 2010.
- [30] Sajad Shirali-Shahreza, Hassan Abolhassani, and M Shirali-Shahreza. Fast and scalable system for automatic artist identification. *Consumer Electronics, IEEE Transactions on*, 2009.
- [31] Li Su and Yi-Hsuan Yang. Sparse modeling for artist identification: Exploiting phase information and vocal separation. In *ISMIR*, 2013.
- [32] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *Speech and Audio Processing, IEEE transactions on*, 2002.
- [33] LJP Van der Maaten, EO Postma, and HJ van den Herik. Matlab toolbox for dimensionality reduction. *MICC*, 2007.
- [34] Rui Xia and Yang Liu. Using i-vector space model for emotion recognition. In *INTERSPEECH*, 2012.

CORRELATING EXTRACTED AND GROUND-TRUTH HARMONIC DATA IN MUSIC RETRIEVAL TASKS

Dylan Freedman

Harvard University

freedmand@post.harvard.edu

Eddie Kohler

Harvard University

kohler@seas.harvard.edu

Hans Tutschku

Harvard University

tutschku@fas.harvard.edu

ABSTRACT

We show that traditional music information retrieval tasks with well-chosen parameters perform similarly using computationally extracted chord annotations and ground-truth annotations. Using a collection of Billboard songs with provided ground-truth chord labels, we use established chord identification algorithms to produce a corresponding extracted chord label dataset. We implement methods to compare chord progressions between two songs on the basis of their optimal local alignment scores. We create a set of chord progression comparison parameters defined by chord distance metrics, gap costs, and normalization measures and run a black-box global optimization algorithm to stochastically search for the best parameter set to maximize the rank correlation for two harmonic retrieval tasks across the ground-truth and extracted chord Billboard datasets. The first task evaluates chord progression similarity between all pairwise combinations of songs, separately ranks results for ground-truth and extracted chord labels, and returns a rank correlation coefficient. The second task queries the set of songs with fabricated chord progressions, ranks each query’s results across ground-truth and extracted chord labels, and returns rank correlations. The end results suggest that practical retrieval systems can be constructed to work effectively without the guide of human ground-truthing.

1. INTRODUCTION

Computational algorithms to approximate harmonic content in a song typically output sequences of chord symbols which can be evaluated in terms of accuracy using their recall compared to human-annotated chord progressions. Leading algorithms to extract chord progressions from audio files have an accuracy of around 80% using popular Western music [12, 15]. Though these algorithms can effectively match human chord-labeling intuitions, it is largely unexplored how these approximated chord annotations perform in typical music retrieval tasks relative to human annotations. In this paper, we propose a method

for evaluating the correlation of music retrieval task results across extracted and ground-truth datasets corresponding to the same collection of songs. We limit the scope of our exploration to chord labels and a few established similarity methods, but the resulting procedure can be generalized to other musical features such as melody, rhythm, and mid-level representations.

1.1 Contribution

This paper explores an alternative way to evaluate the efficacy of algorithms to extract musical features from songs. Rather than simply calculate accuracy of computationally extracted information relative to a reference, or ground-truth, dataset, we propose the use of *correlational metrics*. Given a set of common music informatics retrieval tasks on a set of songs, correlational metrics quantify to what extent the output results differ between two input sets: computationally extracted and ground-truth features for the same set of songs. Testing this system on a chord labeling algorithm, we design an alignment-based system to calculate harmonic similarity, devise two simple tasks—evaluating similarity between pairs of songs and querying by chord progression—and use a global optimization algorithm over the system’s parameters to maximize the resulting correlational metric. The input datasets used and the design of the system are described in the following sections.

2. CHORD PROGRESSION DATASETS

The selection of songs we consider in this paper is motivated by availability. In order to compare ground-truth and computationally extracted chord datasets, it is necessary to have a set of song files, their corresponding ground-truth chord progression data, and a computer algorithm to extract chords from the audio files and create an extracted chord dataset. The number of reliable research-backed human ground-truth chord progression datasets is scarce, thus to maintain a separation of algorithm from data, it is useful to use a chord extraction algorithm that predates the ground-truth dataset such that it could not have been trained against any of its data.

2.1 Chord Extraction

*Chordino*¹ is an open-source chord extraction software program written by Matthias Mauch based on his winning

¹<http://isophonics.net/nlms-chroma>



© Dylan Freedman, Eddie Kohler, Hans Tutschku.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Dylan Freedman, Eddie Kohler, Hans Tutschku. “Correlating Extracted and Ground-Truth Harmonic Data in Music Retrieval Tasks”, 16th International Society for Music Information Retrieval Conference, 2015.

2009 and 2010 MIREX chord estimation algorithm submissions [4, 15]. Chordino achieves an 80% chord symbol recall and is still considered state-of-the-art [16]. Though an algorithm by Khadkevich [12] currently has the highest chord symbol recall in the 2014 MIREX audio chord estimation task, there is no publicly released source code for his work, whereas Chordino is available as a *VAMP*² plugin. The ground-truth dataset we use, as detailed in the following subsection, was compiled in 2011. Unlike Khadkevich's chord identification algorithm released in 2014, there is no possibility that Chordino could have been influenced by or tested against this dataset, maintaining a purity of separation between data and system. Chordino is the only chord extraction algorithm considered in this paper and is used with default settings.

2.2 Ground-Truth Dataset

The *McGill Billboard* annotations collected in [3] and freely available online³ are a state-of-the-art human-annotated chord dataset. The dataset is comprised of over 1,000 songs sampled from different decades from the 1950s to the early 1990s across different Billboard charts from the United States "Hot 100".⁴ The researchers hired music experts and professional jazz musicians to annotate the songs randomly sampled from the Billboard charts. Each song was annotated twice to maintain a standard of accuracy. The resulting dataset is the most comprehensive current ground-truth set of chord annotations and is used in recent MIREX chord annotation competitions. Importantly, the dataset postdates the Chordino chord extraction algorithm, obviating the possibility of training bias.

We were able to locate source audio for 529 of the McGill songs. The corresponding ground-truth annotations for these 529 form the *ground-truth McGill dataset*, or *McGill_g*. We extracted chord annotations for each of these 529 songs using Chordino with default settings, leading to the creation of the *extracted McGill dataset*, or *McGill_e*. To maintain a consistent chord alphabet, we simplify the harmonies used within the ground-truth dataset to match the closest chord within the alphabet of chord qualities used by Chordino. We preserve the root and bass notes of each chord and evaluate the closest simplified chord using the *Harte* metric as described in Subsection 3.2.1.

3. A HARMONIC SIMILARITY SYSTEM

3.1 Smith-Waterman Local Alignment Algorithm

The Smith-Waterman algorithm [17] is a dynamic programming algorithm that searches through two sequences exhaustively, looking for the pair of subsequences with optimal similarity based on the cost of transforming one subsequence into the other using three operators. The sequences are composed of symbols within an alphabet Σ . The first operator, *substitution*, defines the cost of transforming any

one symbol into any other and can be represented as a two-dimensional cost matrix S , where $|S| = |\Sigma| \times |\Sigma|$. The second and third operators, *insertion* and *deletion*, quantify the cost of removing or adding a number of elements at a certain position in one of the subsequences, resulting in gaps in the final alignment. These two operators can be represented concisely using a gap function W that assigns costs to gaps of specified lengths. Given a substitution matrix S with a negative expected value but positive values for similar input symbols, the Smith-Waterman algorithm effectively isolates the strongest local regions of similarity corresponding to the highest score.

Smith-Waterman is useful in the context of comparing chord progressions as it has mechanisms to deal well with inexact data, using different gap costs and chord substitution functions that compensate for small errors. To account for songs in different keys, the score that is returned can be the maximum Smith-Waterman score of all twelve transpositions of one sequence relative to the other. Assuming a fixed substitution and gap function, let $sw(s_1, s_2)$ return the Smith-Waterman score for two sequences s_1 and s_2 . If $t(s, i)$ is a transpose function that returns a transposed sequence given an input sequence s and a number of semitones i , we can express our final score as a similarity function SW :

$$SW(s_1, s_2) = \max_{i=0}^{11} sw(s_1, t(s_2, i)) \quad (1)$$

Due to its advantages and research that supports its efficacy [6, 10], the Smith-Waterman algorithm will be used to compare chord progressions in this paper and quantify harmonic similarity. There are downsides to the Smith-Waterman algorithm. In its current form, the score returned reflects only the optimal local alignment and does not consider other strong subregions of similarity. Allali et al. [1] describe a process for constructing a 3-dimensional Smith-Waterman algorithm that can account for modulations to a new key signature mid-song. These adaptations leave room for future experimentation. This paper focuses on only returning one optimal local alignment score in the highest scoring transposition.

3.2 Parameters

We chose a number of parameters to alter the nature of the Smith-Waterman algorithm used. These parameters are used with global optimization techniques to find good settings such that ground-truth and extracted chord annotations perform similarly.

3.2.1 Chord Distance Functions

We consider two chord distance metrics. Like Haas et al. [6], we use Lerdahl's Tonal Pitch Space (*TPS*) [14] as a chord distance function to populate the substitution matrix S . *TPS* quantifies the distance between two chords relative to the key signature of a song based on psychological qualities of human chord perception. We utilize the key finding approach in [6] to establish the tonic and mode of each song we are considering and assume no transpositions occur midsong. We additionally consider a metric

² <http://www.vamp-plugins.org/>

³ <http://ddmal.music.mcgill.ca/billboard>

⁴ <http://www.billboard.com/charts/hot-100>

proposed in Harte’s PhD thesis (Harte) [11] which quantifies the fraction of similar pitch classes between two chords over their cumulative set of pitch classes. If $P_c(c)$ returns the set of pitch classes for a given chord c this can be expressed as:

$$Harte(c_1, c_2) = \frac{|P_c(c_1) \cap P_c(c_2)|}{|P_c(c_1) \cup P_c(c_2)|} \quad (2)$$

We denote a variable C_d to correspond to which distance function is used, *TPS* or *Harte*.

We additionally devise two parameters to scale and subtract the function C_d such that a cost matrix S populated by C_d has a negative expected value. We first normalize the chord distance function to a value in $[0,1]$, where 0 indicates no similarity and 1 perfect similarity. In *TPS* this requires a division by 13. m_x represents the amount by which this normalized value is multiplied and m_s the amount it is subtracted. We round this number to the nearest integer out of consideration for the Smith-Waterman implementation we used. We arbitrarily only considered integers from 1 through 30 inclusive for both m_x and m_s as values in this range seemed to achieve a good resolution of scaled chord distance values. Finally, S can be populated based on the final scaled and subtracted value and choice of C_d by iterating over all possible pairs of chords in Σ .

3.2.2 Gap Costs

We only consider one class of gap functions, *affine gap functions* [9], which can be defined by the following equation for gaps of size $i \geq 1$:

$$W(i) = -gap_{open} - gap_{extension} \cdot (i - 1) \quad (3)$$

The two constants gap_{open} and $gap_{extension}$ are parameters that can be changed to alter the penalty of the initial gap and following gaps in the sequence alignment, a potentially useful feature to model an initial alignment gap being more or less costly than subsequent gaps. In our implementation, we considered integer values ranging from 0 through 127 inclusive for gap_{open} and $gap_{extension}$.

3.2.3 Normalization

A difficulty with comparing Smith-Waterman scores is that they tend to have a positive correlation with increased sequence length. There are approaches to combat this effect using statistical learning techniques [2]. We tested a simpler normalization metric that returns values in $[0,1]$:

$$SW_{norm}(s_1, s_2) = \frac{SW(s_1, s_2)}{\max\{SW(s_1, s_1), SW(s_2, s_2)\}} \quad (4)$$

We devise a parameter CP_d to represent the chord progression similarity function used, SW or SW_{norm} .

4. EXPERIMENTAL DESIGN

This paper tests how similarly common harmonic music retrieval tasks perform using extracted chord data versus

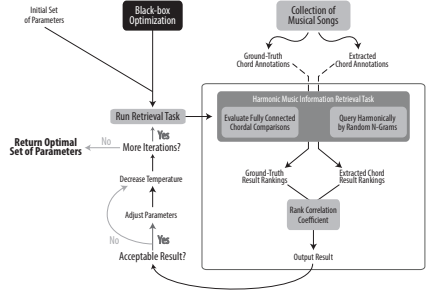


Figure 1. Flowchart of experimental design. This experiment requires a collection of songs with corresponding ground-truth and computationally extracted chord annotations. These different chord datasets describing the same collection of songs are fed into a harmonic retrieval task in isolated experiments, each producing a different result list. These result lists are ranked and correlated to return a correlational metric. Global optimization techniques search for maximum correlational metric scores by running many iterations of the retrieval task with changing parameters based on the performance of the correlational result relative to previous iterations. The returned set of parameters represents an approximate optimal configuration for minimizing algorithmic differences between human and computationally extracted chord inputs.

human-produced data. We primarily test two tasks across $McGill_g$ and $McGill_e$ datasets, rank both sets of results, and calculate a correlational metric P . We then run a black-box optimization strategy to approximate a maximum for this correlational metric across the different parameters detailed in Section 3.2. This process is outlined in a flowchart in Figure 1.

4.1 Retrieval Tasks

This subsection describes the two high-level tasks that form the substance of the experiments. Inputted with the parameters described in the previous section, these algorithms perform chord progression comparisons over a collection of songs using the harmonic similarity system previously outlined to accomplish a common music retrieval objective. The result is a collection of harmonic similarity scores that can be enumerated in an ordered fashion.

4.1.1 Fully Connected Pairwise Harmonic Comparison

This method (*FCC*), given parameters and a collection of chord annotations, returns the harmonic similarity scores for every pairwise combination of songs. The algorithm proceeds in a well-ordered manner such that no pair of songs is iterated twice and results are consistently positioned across two sets of chord annotations corresponding to the same collection of songs (e.g. $McGill_g$ and $McGill_e$).

4.1.2 Query by N-gram

This retrieval task (*QBN*), given parameters and a collection of chord annotations, involves comparing the collection of annotations with random chord sequence queries to simulate a basic search algorithm. Each query sequence is compared with every song in the database, and a two-dimensional table of harmonic similarity scores is returned.

We initially fabricate 100 query sequences, generated randomly within the alphabet of chord qualities Σ but used consistently across experiments and song collections. Out of the 100 query sequences, four groups of 25 query sequences are generated with lengths of 4, 8, 16, and 32, respectively. Each query sequence is padded in length by repeating itself such that the length is at least that of the longest song in the collection so that the Smith-Waterman scores are not restricted by length of query sequence. We chose this repetition of query sequences to imitate the repetitive structure of musical songs and emphasize the cyclic nature of chord progression perception. For each of the 100 query sequences, *QBN* collects harmonic similarity scores by comparing the query sequence against each of the songs in the given collection. The result is a two-dimensional table of harmonic similarity scores of size 100 by the length of the input collection of songs.

4.2 Correlational Metrics

4.2.1 Ranking and the Spearman Correlation Coefficient

The ranking of a sequence is a mapping of every element of the sequence to its position in the sequence. This ranking is done such that elements with the same value are assigned the average index of their positions. Two equally sized lists of rankings s_1 and s_2 can be assigned a correlation coefficient based on the Spearman correlation coefficient (ρ) [7]. If n is the length of one of the ranked lists, ρ can be calculated:

$$\rho(s_1, s_2) = 1 - \frac{6 \sum_i^n (s_{1i} - s_{2i})^2}{n(n^2 - 1)} \quad (5)$$

ρ returns a number in $[-1, 1]$, with 1 indicating a perfect positive correlation, -1 a perfect negative correlation, and 0 no correlation.

4.2.2 Calculating the Correlational Metric

For each of the two experimental tasks, we compare the resulting harmonic similarity scores across ground-truth and extracted chord annotations corresponding to the same set of songs, *McGill_g* and *McGill_e*.

The resulting correlational metric P is calculated by ranking the result lists for *McGill_g* and *McGill_e* separately and returning a rank correlation between both resulting lists. For *FCC*, P is calculated by simply ranking each result list and returning the Spearman correlation coefficient ρ between the two ranked lists. For *QBN*, each result list from *McGill_g* and *McGill_e* for each of the 100 queries generated is ranked independently. The correlation coefficients ρ for each of the 100 pairs of ranked lists is averaged and returned as P .

Variable	Notation	Values
Similarity Function	CP_d	$\{SW, SW_{norm}\}$
Gap Open Cost	gap_{open}	$[0, 128]$
Gap Extension Cost	$gap_{extension}$	$[0, 128]$
Chord Distance	C_d	$\{Harte, TPS\}$
Distance Multiplier	m_x	$[1, 30]$
Distance Subtractor	m_s	$[1, 30]$

Table 1. Summary of experimental parameters.

4.3 Global Optimization with Simulated Annealing

To derive optimal parameters to maximize the correlational metric P across tasks, we use a basic implementation of the simulated annealing algorithm [5, 13]. Let f refer to one of the retrieval tasks that takes as input a set of parameters s_t and runs over *McGill_g* and *McGill_e* to return a correlational metric P .

We try to stochastically search for parameters in s_t to maximize $f(s_t)$. Simulated annealing takes a function, $move(s_t)$, which returns a new state s'_t that is slightly changed from s_t in a random manner. f is recalculated with s'_t to see if the move was beneficial. A temperature variable T stores acceptable deltas between old and new states. If $|f(s'_t) - f(s_t)| > T$, the move is rejected and s_t is left unchanged; otherwise, s_t takes on the new state value, s'_t .

Simulated annealing runs with a fixed number of iterations i_t . In each iteration, we perform $move(s_t)$, and following each iteration, T exponentially decreases. This gives the optimization process more exploratory freedom in initial stages when T is higher. After i_t iterations, the resulting s_t is an approximate maximum of f . This algorithm is useful in search spaces that are sufficiently complex or large, such that exact optimization algorithms are infeasible.

4.3.1 Implementation

Let s_t contain our parameters (see Table 1): $\{CP_d, gap_{open}, gap_{extension}, C_d, m_x, m_s\}$. The $move$ function represents a transition to a nearby state—as each variable in s_t is an integer, the jump must be discrete. Our $move$ implementation takes a random step following a normal distribution for each variable in the state, rounding the result to the nearest integer and ensuring the value falls within the bounds of the variable. The standard deviation of this random step for each variable is chosen to be $\frac{1}{3}$ of that variable's range. CP_d and C_d , taking two possible function values each, can be treated as integer variables with values in $\{0, 1\}$. If a move results in a combination of parameters such that the expected value of S is not negative or there are no positive values, the scaling and subtraction factors m_x and m_s are randomized again from their last values following the same normal distribution jump process. This process repeats until S has a negative expected value and some positive values so the Smith-Waterman algorithm can effectively isolate localized chord comparison results.

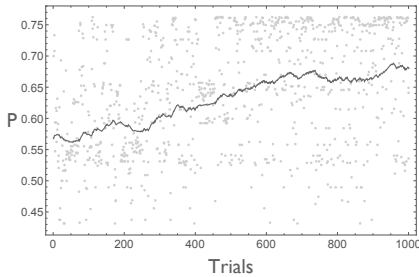


Figure 2. Simulated annealing performance in *FCC*. Each dot represents an iteration of the algorithm and correlational metric P . The jagged line, an exponential moving average, demonstrates the relatively constant increase in performance as iterations progress.

For each task, *FCC* and *QBN*, we run 1,000 iterations of simulated annealing to optimize the correlational metric P with a temperature T that starts at 1 and decreases exponentially to 0.005 at the final iteration.

5. RESULTS

In this section, we detail the results of the optimization procedures across the two retrieval tasks (*FCC* and *QBN*) as detailed in Subsection 4.1.

5.1 Optimizing Fully Connected Comparison

Across 1,000 iterations of simulated annealing for the *FCC* task, the correlational metric P at each iteration generally increased (see Figure 2). The maximal P returned by the simulated annealing was 0.7619, indicating a strong correlation. The parameters s_t resulting in this correlation first occurred at iteration 472 with values $\{CP_d : SW, gap_{open} : 0, gap_{extension} : 28, C_d : TPS, m_x : 5, m_s : 9\}$. The correlation between the ranked result lists for ground-truth and extracted chord data with these parameters can be visualized with a 3-dimensional histogram in Figure 3.

A common measure for accuracy in music retrieval is the Average Dynamic Recall (ADR) [18], which has been used to evaluate similarity assessments in MIREX competitions since 2005. In the context of retrieval results, ADR assesses at all given position how many songs have occurred up to that position that should have occurred relative to ground-truth rankings, returning an average in $[0,1]$, with 1 indicating perfect similarity. We calculated the ADR of the *FCC* results list of extracted chord data relative to the generated ground-truth results list, deriving a result of 0.7664. As a warning, this measure is not particularly applicable to our work as the output ground-truth results list does not demonstrate an actual ground-truth similarity assessment, but its use here nonetheless illustrates the correlation of this parameter set in the context of music retrieval.

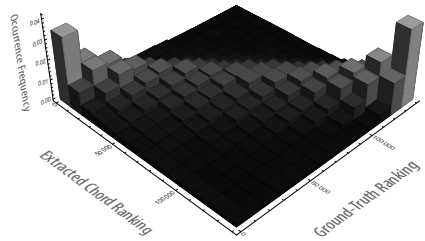


Figure 3. 3-dimensional histogram of the optimal fully connected comparison (*FCC*) rankings. The correlation ($\rho=0.76$) is visible through the elevated diagonal band. The density of points along this band is greatest at the corners as evidenced by bin heights—this means salient strongly and weakly ranked chord progression results are most preserved by the parameters that led to this result.

5.2 Optimizing Query by N-grams

Like *FCC*, the correlational metric P also generally increased across iterations in *QBN* (see Figure 4). The maximal P returned by simulated annealing was 0.7790, occurring singularly with the parameters $s_t = \{CP_d : SW_{norm}, gap_{open} : 1, gap_{extension} : 82, C_d : TPS, m_x : 1, m_s : 10\}$. The average ADR across each of the 100 queries with these parameters was 0.7900.

Task	P	ADR
<i>FCC</i>	0.7619	0.7664
<i>QBN</i>	0.7790	0.7900

Table 2. Summary of experimental results.

5.3 Parameter Optimization

The harmonic retrieval tasks presented in this paper, *FCC* and *QBN*, rely on a common set of parameters s_t . Though generalizations on effective values for the parameter set cannot be fully founded, it can still be useful to future experimentation to detail average correlational metric values associated with ranges of parameter values from the simulated annealing experiments.

CP_d and C_d are the variables that perhaps change the nature of the Smith-Waterman function the most fundamentally. Average output correlational metric values for inputted choices of CP_d and C_d are as follows:

	FCC		QBN	
	<i>Harte</i>	<i>TPS</i>	<i>Harte</i>	<i>TPS</i>
<i>SW</i>	0.59	<u>0.68</u>	0.40	0.49
<i>SW_{norm}</i>	0.56	0.62	0.35	<u>0.53</u>

where maximum values are underlined. According to these observational results, *TPS* outperforms the *Harte* chord distance metric in both experiments in terms of maximizing correlation.

gap_{open} and $gap_{extension}$ take a wider range of values, thus it is more useful to look at variable ranges and their

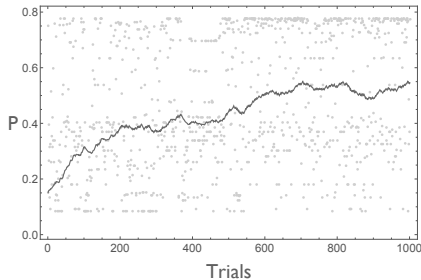


Figure 4. Simulated annealing performance in *QBN*.

average outputs. Following are correlational metrics corresponding to ranges of gap variable values:

Range	FCC		QBN	
	<i>gap_{open}</i>	<i>gap_{extension}</i>	<i>gap_{open}</i>	<i>gap_{extension}</i>
0	0.71	0.64	0.69	0.49
1-8	0.67	0.62	0.46	0.49
> 8	0.61	0.64	0.34	0.47

These results suggest that gap opening penalties of 0 influence higher correlational harmonic metric score.

Finally, we chart which scaling and subtraction factors, m_x and m_s , produced the highest average correlation metric scores:

	FCC			QBN			
	m_x	m_x	m_x	m_x	m_x	m_x	
	1-9	10-19	20+	1-9	10-19	20+	
m_s	1-9	0.67	0.64	0.69	0.62	0.51	0.65
	10-19	0.68	0.65	0.65	0.51	0.43	0.49
	> 20	0.58	0.58	0.57	0.39	0.38	0.30

These results are both consistent in assigning higher correlational metric scores to large multiplication factors and small subtraction factors. A possible explanation for this behavior and favoritism towards gap penalties of 0 is that these factor choices result in the highest Smith-Waterman expected values and result scores. Though this expected value is ensured to be negative, a value close to 0 will more frequently match chords positively by chance and result in longer local alignment scores that resemble global alignment scores. It is possible that global sequence alignment techniques used in *FCC* and *QBN* have strong correlational harmonic metric scores. Further research in global sequence alignment could present promising correlational metric results.

6. DISCUSSION

This paper suggests a new class of similarity assessments in music information retrieval (MIR), *correlational metrics*, and outlines an experimental procedure for assessing these metrics. Correlational metrics capture the degree to

which ground-truth and extracted features perform similarly through retrieval tasks. It is possible that similar results in a retrieval task do not necessarily imply correct or good results. The experimental choices made in this paper, such as using local alignments and the chord distance metrics, are demonstrated in MIR research as strong choices for matching human intuitions of similarity [8, 11]; however, these experimental choices in this paper reflect one possible use case. In the context of chord progressions, there does not exist any reliable ground-truth similarity assessments, which motivated this work.

Further experimentation is necessary with different chord extraction algorithms and settings. The chord extraction algorithm used in this paper is highly accurate, which may imply stronger correlational metric scores. Testing a variety of chord extraction algorithms would render a comparison of correlational metric scores associated with a gradient of extraction algorithm accuracies, giving statistical significance to the resulting scores and potentially uncovering other salient observations. Once there exist research-backed ground-truth similarity assessments for chord progressions, this work can be enriched with direct comparisons to human intuitions. In its current form, this paper is limited to Western harmonies, and more specifically, pop songs from the 1950s onwards. Many other features could be investigated within our experimental design, from those directly supplemental to harmony, such as chord duration and melody, to external factors, such as song popularity or artist. Incorporating and testing more chord distance metrics and different parameters and ranges would additionally benefit this class of research. Modifying the retrieval tasks and implementing additional tasks could extend this work, as well. For instance, randomized query sequences in the *QBN* task could be generated according to probabilistic n-gram models to match more likely search inputs and limit bias in the resulting correlational metric score as a result of purely random queries being unnatural and distant to the input datasets.

Assuming the parameter choices that resulted in the optimal correlational metrics in this paper resulted in a harmonic similarity metric that matches human intuitions of similarity, this paper suggests that effective MIR systems can be constructed without the need for ground-truth chord annotations and provides a framework for conducting such experiments. As there are few research-backed ground-truth chord datasets, this could massively expand the possible realm of chord datasets to reliably harmonically compare. Correlational metrics can also be used in future research across other musical features. The potential implications of this paper suggest that with proper algorithms and parameters that currently exist in the literature, practical MIR systems can be constructed and optimized to work without the guide of human ground-truthing in similarity assessments.

7. REFERENCES

- [1] Julien Allali, Pascal Ferraro, Pierre Hanna, and Costas Iliopoulos. Local transpositions in alignment of poly-

- phonic musical sequences. In *String Processing and Information Retrieval*, pages 26–38. Springer, 2007.
- [2] Eric Breimer and Mark Goldberg. Learning significant alignments: An alternative to normalized local alignment. In *Foundations of Intelligent Systems*, pages 37–45. Springer, 2002.
- [3] John Ashley Burgoyne, Jonathan Wild, and Ichiro Fujinaga. An Expert Ground-Truth Set for Audio Chord Recognition and Music Analysis. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 633–638, 2011.
- [4] Chris Cannam, Matthias Mauch, Matthew EP Davies, Simon Dixon, Christian Landone, Katy Noland, Mark Levy, Massimiliano Zanoni, Dan Stowell, and Luis A Figueira. Mirex 2013 entry: Vamp plugins from the centre for digital music, 2013.
- [5] Vladimír Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45(1):41–51, 1985.
- [6] W Bas De Haas, Matthias Robine, Pierre Hanna, Remco C Veltkamp, and Frans Wiering. Comparing approaches to the similarity of musical chord sequences. In *Exploring Music Contents*, pages 242–258. Springer, 2011.
- [7] Persi Diaconis and Ronald L Graham. Spearman’s footrule as a measure of disarray. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 262–268, 1977.
- [8] Pascal Ferraro and Pierre Hanna. Optimizations of local edition for evaluating similarity between monophonic musical sequences. In *Large Scale Semantic Access to Content (Text, Image, Video, and Sound)*, pages 64–69. Le Centre de Hautes Etudes Internationales d’Informatique Documentaire, 2007.
- [9] Osamu Gotoh. An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, 162(3):705 – 708, 1982.
- [10] Pierre Hanna, Matthias Robine, and Thomas Rocher. An alignment based system for chord sequence retrieval. In *Proceedings of the 9th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 101–104. ACM, 2009.
- [11] Christopher Harte. *Towards automatic extraction of harmony information from music signals*. PhD thesis, Department of Electronic Engineering, Queen Mary, University of London, 2010.
- [12] Maksim Khadkevich and Maurizio Omologo. Time-frequency reassigned features for automatic chord recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference*, pages 181–184. IEEE, 2011.
- [13] Scott Kirkpatrick et al. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.
- [14] Fred Lerdahl. Tonal pitch space. *Music Perception*, pages 315–349, 1988.
- [15] Matthias Mauch and Simon Dixon. Approximate note transcription for the improved identification of difficult chords. In *Proceedings of the 11th International Society for Music Information Retrieval Conference*, pages 135–140, 2010.
- [16] Matt McVicar, Raúl Santos-Rodríguez, Yizhao Ni, and Tjil De Bie. Automatic chord estimation from audio: A review of the state of the art. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 22(2):556–575, 2014.
- [17] Temple F Smith and Michael S Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197, 1981.
- [18] Rainer Typke, Remco C Veltkamp, and Frans Wiering. A measure for evaluating retrieval techniques based on partially ordered ground truth lists. In *Multimedia and Expo, 2006 IEEE International Conference*, pages 1793–1796. IEEE, 2006.

Poster Session 3

CLASSICAL MUSIC ON THE WEB – USER INTERFACES AND DATA REPRESENTATIONS

Martin Gasser¹, Andreas Arzt^{1,2}, Thassilo Gadermaier¹,
Maarten Grachten¹, Gerhard Widmer^{1,2}

`martin.gasser@ofai.at, andreas.arzt@jku.at, thassilo.gadermaier@ofai.at,`
`maarten.grachten@ofai.at, gerhard.widmer@jku.at`

¹Austrian Research Institute for
Artificial Intelligence (OFAD), Vienna, Austria

²Dept. of Computational Perception
Johannes Kepler Universität, Linz, Austria

ABSTRACT

We present a set of web-based user interfaces for explorative analysis and visualization of classical orchestral music and a web API that serves as a backend to those applications; we describe use cases that motivated our developments within the PHENICX project, which promotes a vital interaction between Music Information Retrieval research groups and a world-renowned symphony orchestra.

Furthermore, we describe two real-world applications that involve the work presented here. Firstly, our web applications are used in the editorial stage of a periodically released subscription-based mobile app by the Royal Concertgebouw Orchestra (RCO)¹, which serves as a content-distribution channel for multi-modally enhanced recordings of classical concerts. Secondly, our web API and user interfaces have been successfully used to provide real-time information (such as the score, and explanatory comments from musicologists) to the audience during a live concert of the RCO.

1. INTRODUCTION

The ways we enjoy music have changed significantly over the past decades, not least as a result of the increased use of internet and technology to deliver multimedia content. Services such as iTunes, Spotify, and YouTube offer easy access to vast collections of music, at any time and any place, through tablets and mobile telephones. Such services typically rely on APIs (Application Programming Interface, a set of HTTP-callable URL's or *API endpoints* providing certain data or functionality) to index and stream multimedia content.

These APIs are often exposed (e.g. last.fm, Soundcloud) to third parties for embedding functionalities into new applications. Services and APIs such as the ones mentioned above are generally geared towards a broad audi-

ence, and offer functionality peripheral to music listening, like searching for music, and creating playlists.

As far as the music listening process itself is concerned, average listeners of popular music access music in a linear fashion, i.e., a piece is consumed from the beginning to the end. However, in the world of classical music, we observed very different requirements - there are many cases that benefit from a more content-oriented infrastructure for delivering music.

We argue there are two important characteristics of classical music that call for a more elaborate treatment of the musical content. First of all, classical pieces tend to be longer and typically have a more elaborate and complex structure than pop songs. Consequently, part of the appraisal of classical music tends to lie in the awareness, and interpretation of that structure, both by musicologists and by listeners. Secondly, as opposed to pop music, in classical music the roles of composing and performing the music are usually clearly separated. This distinction leads to a stronger notion of *piece* on the one hand, and *performance* on the other.

The desire to gain insight into structural aspects of the piece and its performance can be formulated as a use case for a general interested audience, which we will call *overseeing the music*. A second use case, *comparing performances*, is centered around the question how different performances may embody different interpretations of the same piece. This use case may be more pertinent to musicologists, or musicians who wish to prepare their performance of a piece. In the *virtual concert guide* use case, audience members are provided with multi-modal information about the music during a concert. As all efforts towards providing a digital concert experience require considerable *editorial support* by experts behind the scenes, we explicitly consider this use case as well. See [8] for a more detailed description and some initial user feedback justifying those use cases.

It is clear that serving these use cases leads to requirements on the service infrastructure that go beyond mere streaming of the data. Most importantly, dealing transparently with synchronized multi-modal information sources including video, audio, musical scores, and structural annotations and visualizations, requires these sources to be

¹<http://www.concertgebouworkest.nl/en/rco-editions/>



© Martin Gasser, Andreas Arzt, Thassilo Gadermaier, Maarten Grachten, Gerhard Widmer

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Martin Gasser, Andreas Arzt, Thassilo Gadermaier, Maarten Grachten, Gerhard Widmer. "CLASSICAL MUSIC ON THE WEB – USER INTERFACES AND DATA REPRESENTATIONS", 16th International Society for Music Information Retrieval Conference, 2015.

aligned to a common timeline, the musical time. In this paper, we present an API for dealing with multi-modal (video, audio, score) data that is geared towards these requirements. Rather than describing the API in detail, we choose to give a brief overview of the entities involved and present various prototype applications that illustrate how this API allows for an in-depth content-oriented presentation of music. In addition to these prototypes, we discuss two real-world applications that rely on this API.

2. RELATED WORK

The general idea of providing multi-modal and content-based access to music has been expressed in a variety of forms in prior work. In [10], Müller et al. present an audio player for multi-modal access to music data. The goal of the *Freischütz Digital* (cf. [11], [12]) project is the development of a multi-modal repository that comprises digitized versions of the libretto, various editions of the musical score, and a large number of audio/video recordings of performances. Dixon et al. demonstrate seamless switching between different versions of the same piece during playback in their MATCH application [3]. Raimond et al. [13] present an extensive framework for publishing and linking music-related data on the web.

As for the symbolic representation of musical scores, MusicXML² is the *de facto* standard format for exchange of digital sheet music, and as such it has largely replaced MIDI³, which is frequently considered an inadequate representation, especially in the field of classical music. Another approach towards a comprehensive representation of western music notation is the Music Encoding Initiative [5]. While we are aware of the advantages of general and flexible frameworks such as Music Ontology [13] and MEI [5], we have settled on a more stripped-down, use case-centric approach that allowed us to reach our goals quickly. We understand that this might mean a redesign of system components at a certain stage, but we believe that an agile approach is beneficial in our case.

In order to be able to process graphic score sheets, we use a custom bar line finding algorithm, since we currently have no need for a complete transcription of the graphical score. See Vigliensson et al. [18] for a description of the problem and the various problems that might occur. For a general discussion of Optical Music Recognition errors and their impact on aligning score to audio, the reader may refer to [16].

3. WEB API

As already mentioned in the introduction, we did not have one single use case in mind. In order to stay as flexible as possible, we decided on implementing a Service Oriented Architecture (SOA)⁴. By explicitly representing the data in the form of HTTP-accessible JSON files, we are able to serve many different applications, either web-based ones

or implemented in the form of native desktop or mobile applications (see section 3 for a brief outline of the functionality currently offered by the web API).

Because we are working with copyrighted material, we had to protect our API (and consequently, also the user interfaces) with an authentication/authorization system that provides different access levels to different users; furthermore, all communication between the front end HTML5 application and the web service API is encrypted.

3.1 Authentication and authorization

Access to all API endpoints is secured via an API key. A special API endpoint is provided that returns an API key as response to submission of username/password credentials. An API key is associated internally with a certain access level that gives the user access to a pre-defined set of resources within the service.

3.2 API resources

The main resources represented in our web service are:

3.2.1 Person

A person can either be a natural person (such as a composer or a conductor) or another acting entity (such as an orchestra).

3.2.2 Piece

A piece is the most general form of a musical composition. A piece references a composer (a person), a set of scores and a set of performances.

3.2.3 Score

A score represents the notated form of a composition. We made the distinction between pieces and scores in order to be able to represent different versions/editions or different orchestrations of the same piece. A score references the corresponding piece and a set of score images.

The score resource also hosts several sub-resources such as score images, a mapping from abstract score position in musical beats to the corresponding graphical position in the score image and information about the position of bar lines and time signature changes in the score. We have also defined a *variant* sub-resource, which represents a derivative version of the score with a certain repetition structure. This is motivated by the fact that the recording of a piece may very well not include all repetitions written in its underlying score, and this is reflected in the actual *score variant* of the recorded performance.

3.2.4 Performance

A performance represents a musical piece performed by a musician or an orchestra. Apart from the actual audio file, the performance resource also contains the alignment information. A score-to-audio alignment provides links between time instants in a symbolic representation of music (such as the beginnings of bars in a score) and corresponding time instants (e.g., the actual note onsets) in a recording

² <http://www.musicxml.com/>

³ <http://www.mid1.org/>

⁴ <http://www.opengroup.org/soa/source-book/soa/soa.htm>

of the performance. Alignments have been created automatically in the first place by the approach described in [4], but they may have been reviewed and corrected by human annotators, in order to increase the accuracy of musical event positions in the audio file.

From the alignment information, it is quite straightforward to compute the musical tempo for each of these events, thus yielding a tempo-curve of the performance. As an alignment can be defined on different granularity levels, such as for each bar or each beat of a bar, an API request can include a parameter that specifies a certain granularity at which the tempo information is to be calculated (e.g., one tempo value per bar or beat). This *tempo* information is also exposed as a sub-resource of *performance* via the API.

Finally, the API provides functionality to calculate perceived loudness values from a given performance recording. In order to calculate loudness information from digital audio signals, we decided on using the LUFS (Loudness Units relative to Full Scale) measure that was introduced in the EBU R128 recommendation [17]. Like the tempo information, loudness values are available at different granularity levels specified in musical time.

4. EXPLORATIVE USER INTERFACES

In this Section, we sketch five different interactive visualizations based on the API described above. We start by discussing some general aspects and design choices of the visualizations.

In our user interface, we strictly follow the concept of *deep linking* [1]. The idea is that if a user wants to discuss an interesting musical passage in a written conversation, she wants to be able to simply send a URL to another user, who can subsequently click on the link, whereupon the receiver sees the user interface in the same state as the sender. Consider the URL `/score/?score=315&variant=40&performance=1328&position=1823.10`, which opens our score viewer interface with a configuration of a score, a score variant, and a performance audio file, and it also jumps directly to beat position 1823.10.

Because of the highly dynamic nature of the user interfaces, we have decided to develop a single page application⁵ that talks directly to our API. In order to simplify development, improve testability of the code, and to enforce modular and reusable development, we use the popular AngularJS⁶ web development framework.

The user interface prototypes are largely inspired by the use cases mentioned in section 1. While the *overseeing the music* use case has been the main motivation behind the hierarchical navigation element (see section 4.1) and the score viewer (see section 4.2), the *comparing performances* use case has led to the development of user interfaces visualizing performance-related parameters of two performances side by side. The *virtual concert guide* use case motivates the integration of a score following and

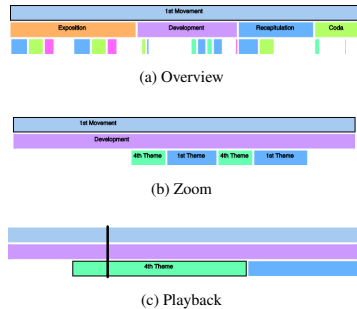


Figure 1: Hierarchical multi-scale navigation in Beethoven's *Eroica*

real-time score display component into a mobile application (see section 5.2), but many components that covered the first two use cases were reused for this use case. Also, in the *editorial support* use case (which is largely covered by the application described in section 5.1), the user interface prototypes have proven useful, as will be discussed below.

4.1 Navigation element

Rather than providing a flat timeline, we propose a navigation element based on a multi-level segmentation of a piece. In his seminal paper [15], Shneiderman laid down some basic principles of how user interfaces for interactive visualization/navigation should be designed: First overview, then zoom and filter, then details on demand (the so-called *Visual Information Seeing Mantra*). Fig. 1 and 2 show how we reflected those principles in our user interface. Fig. 1a shows a three-level visualization: On the top level, we see the name of the musical piece (in this case, the first movement of Beethoven's *Eroica*). The medium level shows the rough structure (Exposition, Development, Recapitulation, Code), and at the lowest level, the position of musical themes is shown. Fig. 1b corresponds to the *zoom* level - the user can use the mouse wheel to literally zoom into the musical structure and study the form of the piece. By dragging the playback cursor to a certain position or clicking on a structural element on any level, the score viewer (see fig. 2) shows the detailed musical notation corresponding to this position. The individual structural elements are also color-coded (this feature can be used to encode repetitions of the same section/theme by using the same color for the visual elements) and the textual annotations appear only on a certain level of detail, in order to prevent text clutter.

See the subsequent sections for a brief description of the *detail* views.

⁵ http://itanat.sourceforge.net/php/spim/api_manifesto_en.php

⁶ <http://angularjs.org>



Figure 2: The score viewer with the interactive structure navigation element on top.

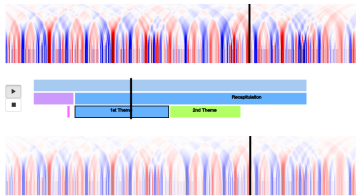


Figure 3: Dynagram visualization of two performances (differences of dynamic levels are shown, where red means increasing, blue means decreasing loudness).

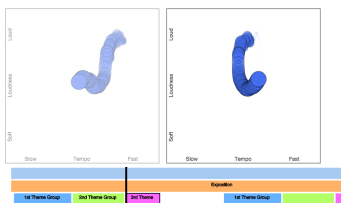


Figure 4: Performance worm visualization of two performances with structure navigation element below.

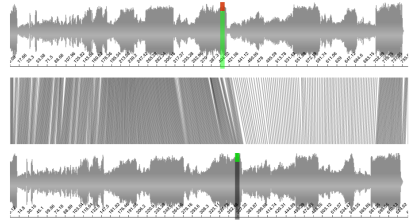


Figure 5: Direct visualization of an alignment

4.2 Score viewer

The score viewer element highlights the bar enclosing the current position (given in musical beats) on the corresponding score sheet. We are currently using scanned and annotated score sheets, but in the future, other score sheet representations (e.g., rendered on demand from MusicXML data) are imaginable and should be evaluated.

4.3 Dynagrams and tempograms

Dynagrams and tempograms show the evolution of a single parameter of the performance (loudness in the case of dynagrams, musical tempo in the case of tempograms) over time, and the parameter is shown on multiple temporal levels. This is achieved by smoothing the parameter-curve over increasing amounts of time and horizontally stacking the results, where the parameter strength is mapped to color. Fig. 3 shows dynagram visualizations of two performances of the same piece that are linked via the navigation element described in 4.1. It integrates short term variations of the respective feature with long term variations into one picture. Therefore, it allows to grasp short term events as well as long term evolution revealing more of the overall structure of the performed piece. This type of visualization builds on earlier work by Langner et al. [7], Sapp [14], and Martorell et al. [9].

We provide two different flavors of these visualizations, (1) where the parameter itself is used as input, showing the absolute values evolving over time and (2) where the derivative of the parameter is used, such that only changes (e.g. crescendo, decrescendo in case of loudness) become visible, as is shown in fig. 3.

4.4 Performance worm

The performance worm is a visualization metaphor integrating the evolution of tempo and dynamics of a musical performance over time in a two-dimensional tempo-loudness space [6]. Its purpose is to uncover hidden characteristics of shaping a performance and the relations between tempo and loudness that are characteristic of a certain style of interpretation. For a given temporal level, the 2 dimensional tempo-loudness-trajectory is displayed where

older values fade into the background as newer data-points are added on top. Fig. 4 shows performance worm visualizations of two performances that are linked via the structural navigation element below.

4.5 Alignment viewer

The alignment visualization shows the waveform displays of the audio signals of two performances of the same piece, and it connects the respective bar line positions (the downbeats) in the two performances. The resulting line pattern reflects the tempo structures of both pieces, and also how they interrelate.

Fig. 5 shows two performances of the fourth movement of Beethoven’s Eroica (Wilhelm Furtwängler conducting the Berlin Philharmonic Orchestra vs. John Eliot Gardiner conducting the Orchestre Révolutionnaire et Romantique). We can conclude from the visualization the intrinsic tempo structure of the performances; while Furtwängler plays the first part slower and becomes faster in the second part, Gardiner chooses to play faster in the first part and become slower afterwards; in both performances, we can observe a strong ritardando in the middle of the piece (i.e., the tempo slows down dramatically), Furtwängler’s ritardando being stronger than Gardiner’s.

5. APPLICATIONS

We employed our API and our visual interfaces in two concrete applications:

5.1 Application 1: Editorial review for a multi-modal music-publishing app

Our PHENICX project partners develop a mobile app for Apple’s iOS devices, and this app is used as a distribution channel for multi-modally enriched recordings of music played by the RCO. One of their selling points is an animated view of the musical score while the audio/video is played back. Fig. 6 sketches the workflow during the production of an edition of the app, and it demonstrates how our API and the score viewer user interface are involved in this process.

The task at hand is to align a recording of orchestral music to a score representation. This problem is twofold: As the graphical score representation is often not available in machine-readable form⁷, the first problem is to find the graphical positions corresponding to musical events (e.g., notes, barlines, or staves). Methods based on Optical Music Recognition turned out to be usable if high-quality graphical scores are available, but we often deal with scanned images from aged printed or even hand-written scores, where standard OMR results are unsatisfactory – therefore, manual intervention is sometimes necessary in this stage. Secondly, the alignment of a recorded performance to a synthesized performance of a score (we used the implementation from [4]) needs to be checked, and manually corrected where necessary.

⁷ MusicXML does encode some score layout information, but not all

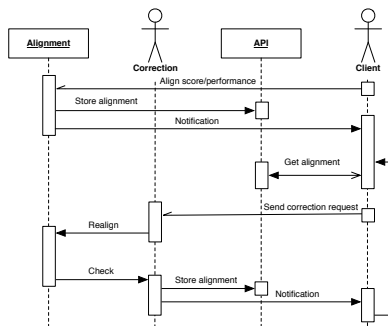


Figure 6: Production/publishing workflow

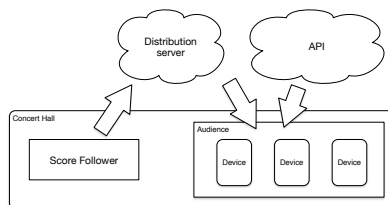


Figure 7: Live score following application

The initial score performance alignment is done by synthesizing the piece from a symbolic representation such as MIDI or MusicXML. After an internal reviewing and correction phase, our client has the opportunity to review the alignment in the score viewer interface and object in case anything is wrong – if this is the case, the alignment goes back into the internal correction phase again. The deep linking functionality – as described in section 4 – has proven to be useful in this iterative process, as it enables the client to pinpoint problematic spots very easily in written conversations. Once the client is satisfied with the quality of the alignment, it is fetched in the form of a JSON⁸ file from our web service API, and further used in the app development process.

5.2 Application 2: Interactive program notes with integrated live score following

The idea is to provide the audience during a concert with additional information about the piece currently played via mobile devices. We have decided to give audience members the possibility to choose between three options, based on personal preference or expertise: (a) an interactive musical score display with the current musical position highlighted, (b) text comments by a musical expert and (c) an artistic video visualization of the music.

⁸ <http://www.json.org>

This application also implies an editorial stage: All three options (a), (b), and (c) rely on sequenced series of events (be it the display of bar positions on a score sheet, timed text messages, or video clips that are played back at certain time instants) that have to be prepared beforehand. In this stage our API and user interfaces are already of use, as editors usually rely on a score representation to pinpoint certain annotations to the music in advance.

During the live event in the concert venue, the client applications, usually running on tablets or smartphones in the audience, access the data (score image sheets, mapping of musical positions to graphical positions) stored in our API. The score follower constantly analyzes the incoming audio stream, and sends the estimated score position to a distribution server. The distribution server subsequently forwards this information to the mobile devices that fetch score images through our API (timed text messages and videos are provided by an additional data source) and use this information to realize an enriched experience for the audience member. Fig. 7 roughly sketches the data flow in this application. We have documented the practicability of our approach in [2].

6. CONCLUSIONS AND FUTURE WORK

We have presented (1) a web service API providing access to structure- and performance-related music data including multimedia elements like score images and audio files and (2) a set of web-based explorative user interface prototypes that act as a frontend to this API. We have also presented two real-life examples of where our API and the user interface prototypes proved to be useful.

We are currently investigating various extensions to our current infrastructure; the workflow described in section 5.1 provides much potential for improvement – we are considering building user interfaces that allow application clients to directly mark alignment errors or even correct alignments directly in the user interface. By fully migrating the alignment service to be based on a complete score representation such as MusicXML instead of scanned score sheets and a MIDI as the corresponding machine-readable score representation, the alignment process would be greatly simplified and the quality of automatic alignments could be improved – in this case, we could reliably identify graphical positions of musical events, and it would be even possible to generate the graphical score directly from the symbolic representation. In addition, more detailed knowledge about instrumentation and performance parameters could also improve the quality of synthesized performances and therefore the quality of resulting automatic alignments (i.e., if a high-quality sample library such as the Vienna Symphonic Library⁹, that allows for precise control of performance parameters, is used).

⁹<https://vsl.co.at/>

7. ACKNOWLEDGMENTS

This research is supported by the European Union Seventh Framework Programme FP7 / 2007-2013, through the PHENICX project (grant agreement no. 601166).

8. REFERENCES

- [1] "Deep Linking" in the World Wide Web. <http://www.w3.org/2001/tag/doc/deeplinking.html>. Accessed: 2015-05-01.
- [2] Andreas Arzt, Harald Frostel, Thassilo Gadermaier, Martin Gasser, Maarten Grachten, and Gerhard Widmer. Artificial Intelligence in the Concertgebouw. pages 165–176, 2015.
- [3] Simon Dixon and Gerhard Widmer. MATCH: A Music Alignment Tool Chest. In *Proceedings of the 6th International Conference for Music Information Retrieval*, number Ismir, pages 492–497, 2005.
- [4] Maarten Grachten, Martin Gasser, Andreas Arzt, and Gerhard Widmer. Automatic Alignment of Music Performances With Structural Differences. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, 2013.
- [5] Andrew Hankinson, Perry Roland, and Ichiro Fujinaga. The Music Encoding Initiative as a Document-Encoding Framework. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 293–298, Miami (Florida), USA, October 24-28 2011.
- [6] Jörg Langner and Werner Goebel. Visualizing Expressive Performance in Tempo–Loudness Space. *Computer Music Journal*, 27(4):69–83, 2003.
- [7] Jörg Langner, Reinhard Kopiez, Christian Stoffel, and Martin Wilz. Realtime analysis of dynamic shaping. In *Proceedings of the 6th International Conference on Music Perception and Cognition*. Keele, UK: Keele University, Department of Psychology, pages 452–455, 2000.
- [8] Cynthia C.S. Liem, Ron van der Sterren, Marcel van Tilburg, Álvaro Sarasúa, Juan J. Bosch, Jordi Janer, Mark Melenhorst, Emilia Gómez, and Alan Hanzalic. Innovating the Classical Music Experience in the PHENICX Project: Use Cases and Initial User Feedback. In *1st International Workshop on Interactive Content Consumption (WSICC) at EuroITV 2013*, Como, Italy, 06/2013 2013.
- [9] Agustín Martorell and Emilia Gómez. Hierarchical multi-scale set-class analysis. *Journal of Mathematics and Music*, 9(1):95–108, 2015.
- [10] Meinard Müller, Frank Kurth, David Damm, and Christian Fremerey. Lyrics-based Audio Retrieval and

- Multimodal Navigation in Music Collections. In *European Conference on Research and Advanced Technology for Digital Libraries*, volume 554975, pages 112–123, 2007.
- [11] Meinard Müller, Thomas Pratzlich, Benjamin Bohl, and Joachim Veit. Freischutz digital: A multimodal scenario for informed music processing. In *Image Analysis for Multimedia Interactive Services (WIAMIS), 2013 14th International Workshop on*, pages 1–4, July 2013.
- [12] Thomas Praetzlich and Meinard Mueller. Freischuetz digital: a case study for reference-based audio segmentation for operas. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, November 4-8 2013.
- [13] Yves Raimond, Samer Abdallah, Mark Sandler, and Frederick Giasson. The Music Ontology. *ISMIR 2007: 8th International Conference on Music Information Retrieval*, 8:417–422, 2007.
- [14] Craig Sapp. *Computational Methods for the Analysis of Musical Structure*. PhD thesis, Stanford University, Department of Music, 2011.
- [15] Ben Shneiderman. The eyes have it: a task by data type taxonomy for information visualizations. *Proceedings 1996 IEEE Symposium on Visual Languages*, 1996.
- [16] Verena Thomas, Christian Fremerey, Meinard Müller, and Michael Clausen. Linking sheet music and audio-Challenges and new approaches. *Dagstuhl Follow-Ups*, 3:1–22, 2012.
- [17] European Broadcasting Union. Loudness Normalisation and Permitted Maximum Level of Audio Signals. <https://tech.ebu.ch/docs/r/r128.pdf>. Accessed: 2015-05-01.
- [18] Gabriel Vigliensoni, Gregory Burlet, and Ichiro Fujinaga. Optical measure recognition in common music notation. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, November 4-8 2013.

A STATISTICAL VIEW ON THE EXPRESSIVE TIMING OF PIANO ROLLED CHORDS

Mutian Fu¹

Guangyu Xia²

Roger Dannenberg²

Larry Wasserman²

¹School of Music, Carnegie Mellon University, USA

²School of Computer Science, Carnegie Mellon University, USA

{mutianf, gxia, rbd, larry}@andrew.cmu.edu

ABSTRACT

Rolled or *arpeggiated* chords are notated chords performed by playing the notes sequentially, usually from lowest to highest in pitch. Arpeggiation is a characteristic of musical expression, or expressive timing, in piano performance. However, very few studies have investigated rolled chord performance. In this paper, we investigate two expressive timing properties of piano rolled chords: *equivalent onset* and *onset span*. Equivalent onset refers to the hidden onset that can functionally replace the onsets of the notes in a chord; onset span refers to the time interval from the first note onset to the last note onset. We ask two research questions. First, what is the equivalent onset of a rolled chord? Second, are the onset spans of different chords interpreted in the same way? The first question is answered by local tempo estimation while the second question is answered by *Analysis of Variance*. Also, we contribute a piano duet dataset for rolled chords analysis and other studies on expressive music performance. The dataset contains three pieces of music, each performed multiple times by different pairs of musicians.

1. INTRODUCTION

Rolled (or *arpeggiated*) chords are notated chords performed by playing the notes sequentially, usually from lowest to highest in pitch. It is a common technique and an integral part of musical expression. Especially, pianists use rolled chords to convey their interpretations of expressive timings. In a very broad sense, every piano chord is rolled since no two notes are played exactly at the same time.

However, very few works have investigated piano rolled chords. As a consequence, when dealing with chords, most *expressive performance* studies stick to the melody or top note, in part due to a lack of theoretical foundations. For example, when analyzing the timing of a chord, researchers usually simply take the onset of a certain note in a chord (e.g., the first note or the highest note) as the onset of a rolled chord [4][13] even though authors realize this is not the best solution. When synthesizing the timing of a chord, people either put the note

onsets of a chord at exactly the same time or decode the onsets of each note individually [6][15]. This situation motivates us to investigate some fundamental properties of rolled chords in order to set a better basis for future expressive performance studies.

We investigate two expressive timing properties of piano rolled chords: *equivalent onset* and *onset span*. Equivalent onset refers to the hidden onset that can functionally replace the onsets of the notes in a chord; onset span refers to the time interval from the first note onset to the last note onset. We compute equivalent onset time and relative location within a rolled chord via local tempo estimation, assuming that local tempo is steady within a few beats. To be more specific, we first estimate a linear mapping (a tempo map) between real performance time and score time for each chord. Then, we compute the intersection between the tempo map and the chord's onset span to compute a hidden equivalent onset. Finally, we compare the equivalent onset with the note onsets of the rolled chord to figure out its relative location. For onset span, we focus on a more fundamental statistical problem: if onset spans are considered random variables, are they drawn from the same distribution, or affected by different chords or performances? We solve this problem by using *Analysis of Variance* (ANOVA). In our case, ANOVA provides a statistical test of whether the means of onset spans of different chords are equal.

The next section presents related work. Section 3 describes a new data set we created for this study. Section 4 presents an important data preprocessing (polyphonic alignment) procedure. In Sections 5 and 6, we show the methodologies for equivalent onset and onset span, respectively. In Section 7, we present experimental results.

2. RELATED WORK

We review two realms of related work: polyphonic alignment and piano rolled chords. The former is only related to our data preprocessing procedure while the latter is related to the main goal of our study.

2.1. Polyphonic Alignment

Researchers have developed both *online* and *offline* polyphonic alignment algorithms for both audio and symbolic data. Our study uses offline symbolic polyphonic alignment based on the MIDI representation.

For audio-based polyphonic alignment, researchers usually first analyze an audio spectrogram to extract pitch and timing features and then perform an alignment



© Mutian Fu, Guangyu Xia, Roger Dannenberg, Larry Wasserman Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Mutian Fu, Guangyu Xia, Roger Dannenberg, Larry Wasserman. "A Statistical View on the Expressive Timing of Piano Rolled Chords.", 16th International Society for Music Information Retrieval Conference, 2015.

based on extracted features. Cont [2] uses non-negative matrix factorization for polyphonic pitch analysis and then uses a hierarchical hidden Markov model to achieve the alignment by sequential modeling. Raphael [11] introduces a graphical method to detect latent tempo and current position in score.

Compared to audio-based approaches, symbolic alignment is relatively easy since the target files usually contain accurate pitch and timing information. Bloch and Dannenberg [1] introduce two online algorithms as a part of the first polyphonic computer accompaniment system. Their work uses pitch information and a rating function to find the best fit between performance and score. Hoshishiba et al. [8] propose an offline approach by using dynamic programming and spline interpolation, in which dynamic programming is used to find the maximum match between performance data and score and spline interpolation is used to post-process and improve the result. A more recent research is done by Chen et al. [3], in which two methods are introduced. The first method sorts notes in a MIDI file by their onset and then uses longest common subsequence to map the performance to the score. The second method sets some correctly matched notes as the pivots, separates note sequence by those pivots, and optimizes the result recursively by forward and backward scanning.

2.2. Piano Rolled Chords Study

There are fewer studies related to piano rolled chords. From an analysis perspective, Repp [12] investigates some descriptive properties of arpeggiated chord onsets by using a single piece of music. To be more specific, this study considers the relative onset timing and inter-onset-interval within arpeggiated chords. It compares the results between the performances by students and experts and draws the conclusion that arpeggiating patterns are subject to large individual differences. From the synthesis perspective, Kim et al. [9] predict the onsets of a rolled chord by first estimating the onset of the highest note and then adding intervals for the onsets of succeeding notes.

3. DATASET

Besides investigating the equivalent onset time and onset span of piano rolled chords, we contribute a piano duet dataset for rolled chord analysis and other studies on expressive music performance [15]. The advantage of duet performance is that we are able to access the expressive timing from both parts. The dataset currently contains three pieces of music: *Danny Boy*, *Serenade* (by Schubert), and *Ashokan Farewell* [7]. Each piece contains a monophonic melody part and a polyphonic accompaniment part. For the polyphonic part, the three pieces contain 32, 56, and 245 chords, respectively. Each piece is performed 35 to 42 times by 5 to 6 different pairs of musicians (each pair performed each piece of music 7

times). This dataset is now accessible online via www.cs.cmu.edu/~gxia/data.

4. DATA PREPROCESSING

Before investigating the equivalent onset and onset span of any rolled chord, we have to align the polyphonic piano performance to the score. This task is done in two steps: *forward alignment* and *backward correction*.

Forward alignment: We adopt the online approach used by Bloch and Dannenberg [1] for the forward alignment step. Generally speaking, the algorithm takes a performance as sequential inputs and matches performance notes one-by-one to a reference of sorted chords. At each step of the alignment, it maximizes the number of matched score notes minus the number of skipped score notes.

Backward correction: The forward alignment procedure works well for most music, but may cause a problem when adjacent chords share the same note.

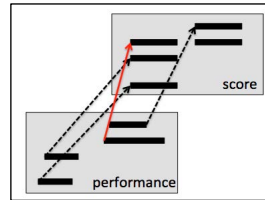


Figure 1. A piano roll illustration of forward alignment procedure.

As shown in Figure 1, dotted arrows represent correct matches while the solid arrow represents the false match. In this case, the top note in the 1st chord is skipped in the performance and the next chord's 1st performed note happens to share the same pitch with the skipped note. As a consequence, the 1st chord "borrows" the missing note from the 2nd chord. In the worst case, if all the chords share the same note, this mismatch behavior could happen recursively. To address this issue, the backward correction algorithm starts from the last chord and recursively recovers the borrowed notes, if any.

5. EQUIVALENT ONSET

If we replace all the note onsets of a rolled chord by a single onset, where should we place this single onset to let it sound most like the original chord? It is reasonable to assume that this equivalent onset is hidden within the range of the rolled chord's onset span and has some particular relationship with the onsets. In this section, we first find out the location of the hidden equivalent onset by local tempo estimation. Then we propose two functional approximations to reveal relative onset location within each rolled chord. In the following sections, we

use n to denote the total number of chords of a piece of music and m to denote the total number of performances of a piece of music.

5.1. Absolute Location of Equivalent Onset

If local tempo around rolled chords is stable, equivalent onsets can be linearly interpolated from neighboring onsets. We consider the melody notes within 2 beats of rolled chords and transfer the equivalent onset estimation problem into a beat estimation problem.

Formally, if the current chord index is i , we denote its score onset and equivalent performance onset by $accom_{s_i}$ and \overline{accom}_{p_i} , respectively. We do equivalent onset estimation based on the melody notes whose onsets are within the range of $[accom_{s_i} - 2, accom_{s_i} + 2]$. To be more specific, we first estimate a linear mapping between performance onsets and score onsets of the melody notes within this range. Then, if we denote the slope and the intercept of this linear mapping as α and β , respectively, we can find the equivalent onset by:

$$\overline{accom}_{p_i} = \alpha \cdot accom_{s_i} + \beta \quad (1)$$

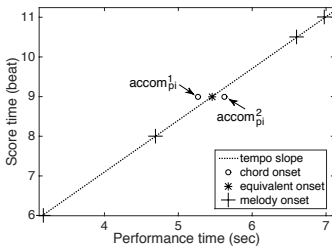


Figure 2. An illustration of equivalent onset estimation by local linear mapping.

This process is illustrated by Figure 2, in which the '+' symbols represent the melody notes and the circle symbols represent accompaniment rolled chord. The line represents the tempo map computed by linear mapping and the star point, on the line at score time 9, represents the equivalent onset computed by equation (1).

5.2. Relative Location of Equivalent Onset

Once the absolute location of equivalent onset is estimated, we present two methods to model its relative location within rolled chords: the ratio model and the constant offset model. For both models, we consider the \overline{accom}_{p_i} computed in the last section as the ground truth and find the models' parameters by minimizing the difference between the models' predictions and the ground truth.

5.2.1 Ratio Model

The ratio model assumes that equivalent onset is decided by the first and last onset of a rolled chord as in the following equation:

$$\overline{accom}_{p_i}'(r) = (1 - r) \cdot accom_{p_i}^1 + r \cdot accom_{p_i}^2 \quad (2)$$

In equation (2), $accom_{p_i}^1$ and $accom_{p_i}^2$ refer to the first and last note onsets in a rolled chord respectively. r is the parameter that characterizes the relative location of equivalent onset. According to the value of r , the equivalent onset can be located as follows:

$r < 0$: equivalent onset is before the first onset of the rolled chord.

$0 \leq r \leq 1$: equivalent onset is between first onset and the last onset of the rolled chord.

$r > 1$: equivalent onset is after the last onset of the rolled chord.

For each piece of music, total number of chords is n and total number of performances is m , we find the optimal r value by equation (3):

$$\hat{r} = \underset{r}{\operatorname{argmin}} \sum_{j=1}^m \sum_{i=1}^n |\overline{accom}_{p_i} - \overline{accom}_{p_i}'(r)| \quad (3)$$

5.2.2 Constant Offset Model

The constant offset model assumes that the equivalent onset is decided by the first onset plus some constant offset s . Formally,

$$\overline{accom}_{p_i}'(s) = accom_{p_i}^1 + s \quad (4)$$

Similar to ratio model, we find the optimal s value by

$$\hat{s} = \underset{s}{\operatorname{argmin}} \sum_{j=1}^m \sum_{i=1}^n |\overline{accom}_{p_i} - \overline{accom}_{p_i}'(s)| \quad (5)$$

6. ONSET SPAN

For onset span, we focus on a more fundamental statistical problem: Do pianists make different interpretations for different chords or performances? As random variables, are all onset spans drawn from the same distribution, or are there different distributions for different chords or performances? In this section, we answer this question by using *Analysis of Variance* (ANOVA). We begin by introducing the basic idea of ANOVA and then link it with our problem step by step.

6.1. One-way ANOVA for Chord Effect

One-way ANOVA can provide a statistical test of whether the means of several groups of data are identical [14]. Formally, if there are n groups indexed by i and μ_i denotes the mean of group i , the null hypothesis and the alternative hypothesis are:

$$H_0: \mu_0 = \mu_1 = \dots = \mu_n \quad (6)$$

$$H_1: \exists i, i': \mu_i \neq \mu_{i'} \quad (7)$$

Generally speaking, one-way ANOVA computes an F-test statistic, which is the ratio of variance between groups to the variance within groups. If different group means are close to each other, this F-test statistic will have a relatively low value and hence retain the null hypothesis.

pothesis. On the other hand, if this F-test statistics is greater than a certain threshold, the null hypothesis will be rejected.

Now let us link this setting to our problem. When checking whether the onset spans of different chords are drawn from the same distribution, each “group” corresponds to a chord and the group members correspond to the onset spans of a particular chord in different performances. In Figure 3, we can see the distributions of the onset span for each chord in *Danny Boy*. The goal is to test whether or not the means of the bars in the boxplot are equal to each other.

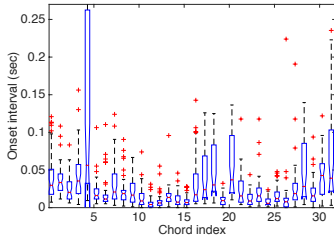


Figure 3. A boxplot of the onset spans of the chords in *Danny Boy*.

Remember each piece of music has n chords and m performances. Therefore, each piece has $N = m \cdot n$ total samples. Referring to the notations in Section 5, the onset span of a rolled chord can be expressed via:

$$t_i = \text{accom}_{p_i}^2 - \text{accom}_{p_i}^1 \quad (8)$$

We use t_{ij} to denote its value in the j^{th} performance. Therefore, the group mean in equation (8) can be computed by

$$\mu_i = \bar{t}_i = \frac{\sum_{j=1}^m t_{ij}}{m} \quad (9)$$

The implementation of one-way ANOVA can be described in the following steps.

First, compute the variation *between* the groups and record its degree of freedom.

$$SS_{btwn} = \sum_{i=1}^n [\bar{t}_i - \bar{t}_{ij}]^2 \quad (10)$$

where $\bar{t}_i = \frac{\sum_{j=1}^m t_{ij}}{m}$, $\bar{t}_{ij} = \frac{\sum_{i=1}^n \sum_{j=1}^m t_{ij}}{n}$. The degree of freedom of SS_{btwn} , $df_{btwn} = n - 1$.

Second, compute the variation *within* individual samples and record its degree of freedom,

$$SS_{within} = \sum_{i=1}^n \sum_{j=1}^m t_{ij}^2 - \sum_{i=1}^n \frac{(\sum_{j=1}^m t_{ij})^2}{m} \quad (11)$$

The degree of freedom of SS_{within} , $df_{within} = N - n$.

Third, compute the F-test statistics by:

$$MS_{btwn} = \frac{SS_{btwn}}{df_{btwn}} \quad (12)$$

$$MS_{within} = \frac{SS_{within}}{df_{within}} \quad (13)$$

$$F = \frac{MS_{btwn}}{MS_{within}} \quad (14)$$

Finally, compare this F-test statistic against a certain threshold to decide whether or not reject the null hypothesis.

6.2. Repeated-measurement One-way ANOVA for Chord Effect

The previous section considered whether different chords have different onset spans. However, an important assumption when using one-way ANOVA is that samples from different groups are independent. In our case, each piece of music is performed by 5 or 6 different pairs of students. Chords played by the same person are clearly correlated. To eliminate the dependent factors produced by same performers, we use repeated-measurement ANOVA to adjust our results.

The general logic of repeated-measurements ANOVA is similar to independent one-way ANOVA. The difference between those two methods is that repeated-measurements ANOVA removes variability due to the individual differences from the within group variance. This process can be understood as removing between-sample variability, and only keeping the variability of how the sample reacts to different conditions (chords). We point readers to Ellen and Girden's book [5] for more detailed descriptions.

6.3. ANOVA for Performance Effect

Section 6.1 and 6.2 presented the method to inspect whether pianists make different interpretations on onset span for different chords. Following a very similar procedure, if we just exchange the index of i and j in 6.1 and keep everything else the same, we can inspect whether onset spans are interpreted differently for different performances.

7. EXPERIMENTAL RESULTS

7.1. Equivalent Onset

7.1.1 Ratio Model

Figure 4 shows the results of the ratio model. In the figure, the x -axis represents the ratio parameter r and the y -axis represents the relative difference (residual) between model estimated equivalent onset and the ground truth computed via local tempo estimation. Therefore, small numbers indicate better results. Each line corresponds to a piece of music. We see that the optimal r values are all within the range from 0 to 1, indicating that the equivalent onset consistently lies within the range of note on-

sets. The optimal values are 0.42 for *Danny Boy*, 0.13 for *Ashokan Farewell*, and 0.78 for *Serenade*.

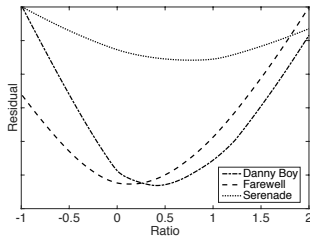


Figure 4. Result of the ratio model.

7.1.2 Constant Offset Model

Similar to Figure 4, Figure 5 shows the results of the constant offset model. The only difference is that the x-axis now represents the constant offset parameter s . We see that the optimal s values are all within the range from 0 to 20 milliseconds. The optimal values are 16 milliseconds for *Danny Boy*, 1 millisecond for *Ashokan Farewell*, and 17 milliseconds for *Serenade*. Compared to the ratio model, the optimal value for constant offset model is more consistent.

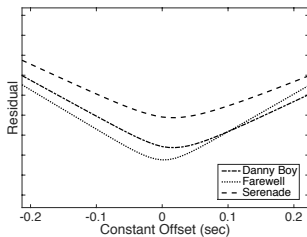
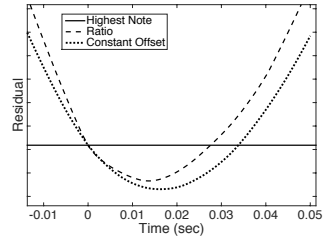


Figure 5. Result of the constant onset model.

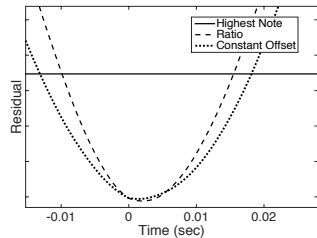
7.1.3 Comparison with Highest Note Model

In most expressive performance studies, people use the highest note onset as the equivalent onset, which we refer to as the “highest note model.” In this section, we compare the results of the ratio model and constant offset model with the highest note model.

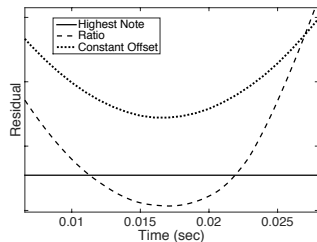
Figure 6 shows this comparison between different models, in which each sub-graph represents a piece of music. Again, smaller number means better prediction. Here, we also map the x-axis value of the ratio model to seconds by multiplying the ratios by the average onset spans. We see that for all the pieces, the ratio model gives better predictions than the highest note model. The constant offset model also does a good job on *Danny Boy* and *Ashokan Farewell* but does not outperform the highest note model for *Serenade*.



(a) Model comparison: *Danny Boy*.



(b) Model comparison: *Ashokan Farewell*.



(c) Model comparison: *Serenade*.

Figure 6. Model comparison of three songs.

7.2. Onset Span

For onset span experiments, we just show the one-way ANOVA table since the repeated-measurement adjustments call for extra notations but give us the same conclusions. Table 1 shows the result of the one-way ANOVA on different chords of *Danny Boy*. Similar to the result of *Danny Boy*, *Ashokan Farewell* and *Serenade* all have the F-test statistics much larger than the thresholds. This indicates that differences between group means are significant. Therefore, we see that not all chords are drawn from the same distribution. In other words, musicians make different interpretations for onset spans of different chords.

Variable	SS	df	F	p
Between	1.6762	31	7.98	4.29×10^{-32}
Within	8.8879	1312		

Table 1. ANOVA for chord effect.

Table 2 shows the result of the one-way ANOVA on different performances of *Danny Boy*. Again, we get similar results for *Ashokan Farewell* and *Serenade*, which all have a F-test statistic not big enough to reject the null hypothesis. This indicates that the differences between group means are *not* significant. Therefore, we see that the interpretations for the same chord’s onset span across different performances are relatively consistent.

Variable	SS	df	F	p
Between	0.2752	41	0.85	0.7383
Within	10.289	1302		

Table 2. ANOVA for performance effect.

8. CONCLUSION AND FUTURE WORK

In conclusion, we create a database to investigate two expressive timing properties of rolled chords in order to set a theoretical basis for future expressive performance studies. We examined three models to characterize the relative location of equivalent onset within rolled chords. The ratio model outperforms the other models for all pieces of music including the highest pitch model used in most research. We also studied onset span. We see that differences are not merely random; musicians use different interpretations for different chords and the interpretation for the same chord across different performances are relatively consistent.

This suggests that in future expressive performance studies, in order to synthesize a rolled chord properly, we can use the equivalent onset as the anchor point (instead of the onset of the highest pitch) and consider the onset span as an important parameter. Although our ratio model improves upon the highest pitch model, the best ratio is different for different pieces and the absolute location of equivalent onset is still based on estimation. This suggests that in future work we should either look for a way to predict the ratio for a given piece of music, or more likely, that we should look for an even better model by combining objective and subjective evaluations.

9. REFERENCES

[1] J. Bloch and R. Dannenberg, “Real-time Computer Accompaniment of Keyboard Performances,” *Proceedings of the International Computer Music Conference*, pp. 279-290, 1985.
 [2] A. Cont, “Realtime Audio to Score Alignment for Polyphonic Music Instruments, Using Sparse Non-negative Constraints and Hierarchical HMMs,” *Proceedings of the International Conference on*

Acoustics, Speech and Signal Processing, pp. 245-248, 2006.
 [3] C. Chen, J. Jang and W. Liou, “Improved Score-Performance Alignment Algorithms on Polyphonic Music,” *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, pp. 1365-1369, 2014.
 [4] P. Desain and H. Honing, “Does Expressive Timing in Music Performance Scale Proportionally with Tempo?” *Psychological Research*, pp. 285-292, 1994.
 [5] R. Ellen and E. Girden, *ANOVA: Repeated Measures*, Sage Publications, 1992.
 [6] S. Flossmann, M. Grachten and G. Widmer, “Expressive Performance Rendering With Probabilistic Models,” *Guide to Computing for Expressive Music Performance*, pp. 75-98, 2012.
 [7] J. Galway and P. Coulter, *Lengends*, Hal Leonard, 1997.
 [8] T. Hoshishiba, S. Horiguchi and I. Fujinaga, “Study of Expression and Individuality in Music Performance Using Normative Data Derived from MIDI Recordings of Piano Music,” *Proceedings of the International Conference on Music Perception and Cognition*, pp. 465-470, 1996.
 [9] T. Kim, F. Satoru, N. Takuya, and S. Shigeki, “Polyhymnia: An Automatic Piano Performance System With Statistical Modeling of Polyphonic Expression and Musical Symbol Interpretation,” *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 96-99, 2011.
 [10] A. Kirke and E. Miranda, *Guide to Computing for Expressive Music Performance*, Springer Science & Business Media, 2012.
 [11] C. Raphael, “A Hybrid Graphical Model for Aligning Polyphonic Audio with Musical Scores,” *Proceedings of the International Conference on Music Information Retrieval*, pp. 387-394, 2004.
 [12] B. Repp, “Some Observations on Pianists’ Timing of Arpeggiated Chords,” *Psychology of Music*, pp. 133-148, 1997.
 [13] B. Repp, “Relational Invariance of Expressive Microstructure across Global Tempo Changes in Music Performance: An Exploratory Study,” *Psychological Research*, pp. 269-284, 1994.
 [14] B. Tabachnick and L. Fidell, *Using Multivariate Statistics*, Haper and Row, 2001.
 [15] G. Xia and R. Dannenberg, “Duet Interaction: Learning Musicianship for Automatic Accompaniment,” *Proceedings of the International Conference on New Interface for Musical Expression*, 2015.

HYBRID LONG- AND SHORT-TERM MODELS OF FOLK MELODIES

Srikanth Cherla^{1,2}

Son N. Tran²

Tillman Weyde^{1,2}

Artur d'Avila Garcez²

¹Music Informatics Research Group, Department of Computer Science, City University London

²Machine Learning Group, Department of Computer Science, City University London

{srikanth.cherla.1, son.tran.1, t.e.veyde, a.garcez}@city.ac.uk

ABSTRACT

In this paper, we present the results of a study on dynamic models for predicting sequences of musical pitch in melodies. Such models predict a probability distribution over the possible values of the next pitch in a sequence, which is obtained by combining the prediction of two components (1) a long-term model (LTM) learned offline on a corpus of melodies, as well as (2) a short-term model (STM) which incorporates context-specific information available during prediction. Both the LTM and the STM learn regularities in pitch sequences solely from data. The models are combined in an ensemble, wherein they are weighted by the relative entropies of their respective predictions. Going by previous work that demonstrates the success of Connectionist LTM, we employ the recently proposed Recurrent Temporal Discriminative Restricted Boltzmann Machine (RTDRBM) as the LTM here. While it is indeed possible for the same model to also serve as an STM, our experiments showed that n -gram models tended to learn faster than the RTDRBM in an online setting and that the hybrid of an RTDRBM LTM and an n -gram STM gives us the best predictive performance yet on a corpus of monophonic chorale and folk melodies.

1. INTRODUCTION

In the present work, our interest is in learning a model to predict a probability distribution over the possible values of the pitch of a musical note in a melody given the sequence of notes leading up to it. The motivation for this stems from theoretical work in musicology and music cognition which attempts to explain various musical phenomena (such as style, genre and mood) in terms of patterns of fulfilment, prolongation and violation of musical expectation [10, 15, 19], i.e., that our perception of music is influenced by how its evolution in time conforms to, or deviates from our expectations. There exists empirical evidence suggesting that these expectations are shaped by an underlying mechanism of statistical learning [9], the consequences of which have also been observed in language

[24]. This apparent commonality between the two domains has inspired the adoption of statistical models for word sequences in language and character sequences in text, to pitch sequences in melody [4, 6, 21, 31]. Previous work interpreting information theoretic concepts such as entropy and mutual information (which play a key role in language and text modelling) in the context of music [5, 16] contributed towards the adoption of these quantities in evaluating such *melody models*. Time-varying *entropy profiles* of predictions made by such models on musical pieces have been used for explaining stylistic implications of salient musical structures [7]. They have also been used to generate melodic stimuli in music cognition research [20]. Predictive models of music have also been used as Music Language Models in music transcription [26]. The reader is referred to [23] for a recent review on predictive machine learning models used in music research.

The melody models considered here contain two components - a long-term model (LTM), and a short-term model (STM) [6]. The parameters of each model are learned through exposure to appropriate data. From a machine learning perspective, the LTM is a model whose parameters are learned offline from a dataset of melodies. It represents more global stylistic characteristics acquired by a listener over a longer time-span. The parameters of the STM are learned online while making predictions on the test data, without any sequence learning occurring in it beforehand. The STM highlights the importance of context-specific information, available in a melody while it is being processed by the listener, in the generation of expectations. Predictions (in the form of probability distributions) made by each model about a certain musical event in a sequence are combined using ensemble methods, and this has been shown to improve the quality of predictions over individual models in the past [6, 21]. The idea of combining corpus-based long-term and context-sensitive short-term predictions from different models was originally a feature of cache-based language models [12]. It was introduced in the context of music in [6], further extended in [21], and adopted in [7, 31].

To address the prediction task, we employ a recently proposed Connectionist model known as the Recurrent Temporal Discriminative Restricted Boltzmann Machine [3]. This model has been shown to have a predictive performance better than n -gram models and other standard Connectionist models on a corpus of monophonic melodies when used as an LTM. We begin by evaluating



© Srikanth Cherla, Son N. Tran, Tillman Weyde, Artur d'Avila Garcez. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Srikanth Cherla, Son N. Tran, Tillman Weyde, Artur d'Avila Garcez. "Hybrid Long- and Short-Term Models of Folk Melodies", 16th International Society for Music Information Retrieval Conference, 2015.

its utility as an STM by carrying out online learning in it, which has not been done previously. Experiments revealed that, while learning did indeed take place, it did not progress quickly enough (as a function of the number of data-points presented to the RTDRBM) to outperform existing state-of-the-art dynamic models based purely on n -grams [22]. On adopting the wisdom of previous work which demonstrated that n -gram models are indeed an effective choice as STMs, we found here that a hybrid prediction model which combines the predictions of an RTDRBM LTM and an n -gram STM achieves better predictive performance, and this also outperforms the state-of-the-art, purely n -gram based dynamic melody models on a corpus of 8 melody datasets. In this paper, we present the results of various LTM-STM combinations that we experimented with to arrive at this result and discuss our observations.

In the next section we formally introduce the task of melody modelling, and entropy-weighted combination strategies for LTMs and STMs. This is followed by a brief overview of the two types of prediction models involved in the present work, in Section 3. Various experiments in combining these models that led to the above mentioned optimal predictive performance are described in Section 4, followed by the conclusions in Section 5.

2. MELODY MODELLING

Our interest is in modelling musical pitch sequences through prediction. The task of music prediction addressed here has strong parallels with previous work in language modelling [14]. Thus, the analogy to natural language is used here to explain it. In statistical language modelling, the goal is to build a model that can estimate the joint probability distribution of subsequences of words occurring in a language L . A statistical language model (SLM) can be represented by the conditional probability of the next word $w^{(T)}$ given all the previous ones $[w^{(1)}, \dots, w^{(T-1)}]$ (written $w^{(1:T-1)}$), as

$$P(w^{(1:T)}) = \prod_{t=1}^T P(w^{(t)} | w^{(1:t-1)}). \quad (1)$$

The present work treats notes in a monophonic melody analogous to words in the above language example. This is inspired by [6] where a similar analogy was made between sequences of characters in the English language and notes in music. We use an event-based representation of music, where the occurrence of each note is treated as a *musical event*. Much in the same way as an SLM, a system for music prediction models the conditional distribution $P(s^{(t)} | s^{(1:t-1)})$ given a sequence $s^{(1:T)}$ of musical events [4, 6, 22] from a musical language S , such that $s^{(t)} \in [S]$, where $[S]$ is the set of symbols (musical pitch values) in S . For each prediction, context information is obtained from the events $s^{(1:t-1)}$ preceding $s^{(t)}$. Although a range of musical features (such as musical pitch, note duration, inter-onset interval, etc.) may be extracted from each musical event as explained in [6], we limit our attention to sequences of musical pitch. And the symbols that

make up these sequences are MIDI values of the pitches which occur in a particular dataset.

2.1 Long- and Short-term Models

In the present work, we make a distinction between two types of prediction models, as introduced previously in the context of *Multiple Viewpoints for Music Prediction* [6]. The first is known as a Long-Term Model (LTM). This model is learned offline on a corpus of melodies (training data), its parameters thus being finalized beforehand and kept constant during the prediction stage. It represents more global stylistic characteristics acquired by a listener over a longer time-span. And the second is what is known as the Short-Term Model (STM). It highlights the importance of context-specific information, available in a melody while it is being processed by the listener, in the generation of expectations. The distinction between the long- and short-term models is also akin to the that made in [11] between “schematic” (LTM) and “veridical” (STM) knowledge in a modular view on music processing. A variant of the LTM which is also considered here was introduced in [22]. This is the LTM+, and in addition to being learned offline on a corpus of melodies like the LTM, it is also updated while making predictions just like the STM. Another distinction between the LTM+ and the STM is that the former is continuously updated across melodies, while the latter is re-initialized after each melody in the test set.

2.2 Combining the LTM & STM

It was demonstrated in [6, 21] that an entropy-weighted combination of the predictions of two or more n -gram models typically results in ensembles with better predictive performance than any of the individual models. As it is the predicted distributions which are combined, this approach is independent of the types of models involved. Here, we briefly describe two rules for creating such ensembles. Let M be a set of models and $P_m(s)$ be the probability assigned to symbol $s \in [S]$ by model m . The first involves taking a weighted arithmetic mean of their respective predictions. This is the *Mean* combination rule, defined as

$$P(s) = \frac{\sum_{m \in M} w_m P_m(s)}{\sum_{m \in M} w_m} \quad (2)$$

where each of the weights w_m depends on the entropy of the distribution generated by the corresponding model m in the combination such that greater entropy (and hence uncertainty) is associated with a lower weight [6]. The weights are given by the expression $w_m = H_{rel}(P_m)^{-b}$, where the relative entropy $H_{rel}(P_m)$ is

$$H_{rel}(P_m) = \begin{cases} H(P_m)/H_{max}(P_m), & \text{if } H_{max}([S]) > 0 \\ 1, & \text{otherwise} \end{cases} \quad (3)$$

The best value of the combination bias $b \geq 0$ is determined through cross-validation. When $b = 0$, all the combined models have the same weight. The quantities H and H_{max}

are respectively the entropy of the prediction and the maximum entropy of predictions over the symbol space $|S|$, and are defined as

$$H(P) = - \sum_{s \in |S|} P(s) \log_2 P(s). \quad (4)$$

$$H_{max}(P) = \log_2 |S|.$$

where $P(X = s)$ is the probability mass function of a random variable X distributed over the discrete alphabet $|S|$ such that the individual probabilities are independent and sum to 1.

The second — the *Product* combination rule, is computed similarly as the weighted geometric mean of the probability distributions. This is given by

$$P(s) = \frac{1}{R} \left(\prod_{m \in M} P_m(s)^{w_m} \right)^{\frac{1}{\sum_{m \in M} w_m}} \quad (5)$$

where R is a normalisation constant which ensures that the resulting distribution over S sums to unity. The weights w_m in this case are obtained in the same manner as in the case of the Mean combination rule. It was observed in a previous application of these two combination methods to melody modelling [21], that the Product rule resulted in a greater improvement in predictive performance.

3. PREDICTION MODELS

Before moving on to the experiments carried out on different LTM-STM combinations in the next section, here we provide a quick overview of the two classes of prediction models that have been employed for this purpose. The first is the Recurrent Temporal Discriminative Restricted Boltzmann Machine, and the other is the n -gram Model.

3.1 Recurrent Temporal Discriminative RBM

The Recurrent Temporal Discriminative Restricted Boltzmann Machine (RTDRBM) [3] was proposed by the authors as the discriminative equivalent of the Recurrent Temporal Restricted Boltzmann Machine (RTRBM) [28]. Both models are identical in structure, and are composed of a sequence of Restricted Boltzmann Machines (RBM) [27], where the visible and hidden layers of the RBM at time-step t are conditioned on the mean-field values of the hidden layer of that at $(t - 1)$ through a set of time-dependent model parameters. The RTDRBM learns the distribution $P(\mathbf{y}^{(1:T)} | \mathbf{x}^{(1:T)})$ over a sequence of input-label pairs $\{\mathbf{x}^{(1:T)}, \mathbf{y}^{(1:T)}\}$, in contrast to the RTRBM which learns the joint probability of the entire sequence $P(\mathbf{y}^{(1:T)}, \mathbf{x}^{(1:T)})$ [1].

The RTDRBM (Figure 1) is obtained by carrying out discriminative learning and inference as put forward in the Discriminative RBM (DRBM) [13], in a temporal setting by incorporating the recurrent structure of the RTRBM which was originally proposed as a generative model for high-

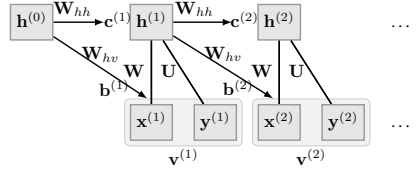


Figure 1: The architecture of the RTDRBM, in which the biases of the visible and hidden layers $\mathbf{b}^{(t)}$ and $\mathbf{c}^{(t)}$ respectively at time-step t are conditioned on the mean-field values of the hidden layer of the RBM $\hat{\mathbf{h}}^{(t-1)}$ at time-step $(t - 1)$. This is also a feature of the RTRBM.

dimensional sequences. This results in the following expression for the posterior probabilities at time-step t :

$$P(\mathbf{y}^{(t)} | \mathbf{x}^{(1:t)}) = P(\mathbf{y}^{(t)} | \mathbf{x}^{(t)}, \hat{\mathbf{h}}^{(t-1)}) \quad (6)$$

It takes into account temporal information carried forward from the previous time-step through the mean-field values of the hidden units $\hat{\mathbf{h}}^{(t-1)}$ [3]. This can be extended to an entire sequence of T events as follows:

$$P(\mathbf{y}^{(1:T)} | \mathbf{x}^{(1:T)}) = \prod_{t=1}^T p(\mathbf{y}^{(t)} | \mathbf{x}^{(t)}, \hat{\mathbf{h}}^{(t-1)}) \quad (7)$$

One can thus learn the model by maximizing the log-likelihood function:

$$\begin{aligned} \mathcal{O} &= \log P(\mathbf{y}^{(1:T)} | \mathbf{x}^{(1:T)}) \\ &= \sum_{t=1}^T \log P(\mathbf{y}^{(t)} | \mathbf{x}^{(t)}, \hat{\mathbf{h}}^{(t-1)}). \end{aligned} \quad (8)$$

Learning here involves updating the model's parameters as dictated by the Backpropagation Through Time (BPTT) algorithm [30]. It was demonstrated in [3] that the RTDRBM outperformed the RTRBM, n -grams and a set of standard Connectionist models on a corpus of 8 different datasets of chorale and folk melodies of varying sizes and complexities when learned offline. In the pitch prediction task of Section 2, the one-hot encoding of the musical event $s^{(t)}$ (which is to be predicted) substitutes the label $\mathbf{y}^{(t)}$ in (6), whereas that of the most recent event from the context $s^{(t-1)}$ substitutes the input $\mathbf{x}^{(t)}$.

3.2 n -gram Model

The n -gram model is a statistical model of sequences that relies on the simplifying assumption that the probability of an event (or in the present case, a musical event) in a sequence depends only on the $(n - 1)$ immediately preceding events [14]. This is known as the Markov assumption, and is applied to model an event sequence $s^{(1:T)}$ as

$$P(s^{(1:T)}) = \prod_{t=1}^T P(s^{(t)} | s^{(t-n+1:t-1)}). \quad (9)$$

where n is known as the order of the n -gram. The model can be represented by a state transition graph, or by a *transition matrix*. Maximum-Likelihood Estimation can be carried out to estimate the parameters of the n -gram model (its transition probabilities) as

$$P(s^{(t)}|s^{(t-n+1:t-1)}) = \frac{N(s^{(t-n+1:t)})}{N(s^{(t-n+1:t-1)})} \quad (10)$$

where $N(s^{(t_1:t_2)})$ is the number of occurrences of a sequence $s^{(t_1:t_2)}$ in the data. As we shall see in Section 4, this simple learning rule is advantageous in an online-learning scenario where the model needs to be constantly updated as it encounters new data. As n -grams rely explicitly on the occurrence frequencies of sequences, it is often the case that the model comes across a never-before-encountered context on which to predict the future event, and this is more common in higher order models. This issue has been dealt with by using *smoothed n -grams* [2] that use lower-order transition probabilities for generating approximations (through interpolation with or scaling of) higher-order probabilities. This also applicable to events that lack a valid context, i.e. $\{s^{(t)} \mid 1 \leq t \leq (n-1)\}$.

The present work employs two of the best variants of the n -gram model evaluated for melody modelling in [22] exclusively as STMs, as an alternative to the RTDRBM which performs poorly in this role (Table 3). Both variants are of unbounded order, wherein they take into account the longest available matching context (of immediately preceding musical events) in order to make a prediction. The first of these (referred to as C^*I) uses the interpolated smoothing method proposed in [18] to account for unfamiliar contexts. The second (referred to as X^*UI) uses a Poisson process based interpolated smoothing method [18] with update exclusion [17]. We refer the interested reader to [22] for further details on these two models.

4. EXPERIMENTAL RESULTS

We evaluate six different LTM-STM combinations. These are listed in Table 1. Also, C^*I and X^*UI are the names

(a) LTM: RTDRBM	STM: n -gram (X^*UI)
(b) LTM: RTDRBM	STM: n -gram (C^*I)
(c) LTM: RTDRBM	STM: RTDRBM
(d) LTM+: RTDRBM	STM: n -gram (X^*UI)
(e) LTM+: RTDRBM	STM: n -gram (C^*I)
(f) LTM+: RTDRBM	STM: RTDRBM

Table 1: Various LTM-STM combinations evaluated here.

of the two best STMs evaluated in a previous study of n -gram based melody models [22].

Each of the combined models was evaluated on 8 melody datasets of different sizes and styles. Prediction cross-entropy was used as the evaluation measure. It was found that combination (b) had the best predictive performance. Furthermore, each case involving an LTM was

Dataset	No. events	$ X $
Yugoslavian folk songs	2691	25
Alsatian folk songs	4496	32
Swiss folk songs	4586	34
Austrian folk songs	5306	35
German folk songs	8393	27
Canadian folk songs	8553	25
Chorale melodies	9227	21
Chinese folk songs	11056	41

Table 2: Melody datasets used for evaluation with their respective total number of musical events and number of prediction categories.

better than its LTM+ counterpart. And finally, the n -grams consistently proved to be a better choice than the RTDRBM as STMs when combined with the same LTM.

4.1 Data

Evaluation was carried out on a corpus of 8 datasets of monophonic MIDI melodies from the Essen Folk Song Collection¹ [25]. The corpus covers a range of musical styles and was previously used in [4, 22] to evaluate their respective prediction models. It contains folk melodies of 7 different traditions, and chorale melodies (Table 2). All melodies are encoded in the ***kern* format in each dataset, and were parsed using the *Music21* Python library [8]. Musical pitch, which occurs as sequences of integer values, is treated as a discrete random variable X , which can assume any of $|X|$ distinct values (or prediction categories).

4.2 Evaluation Measure

Given that the models predict a probability distribution over X at every time-step, their goal may be viewed as one of minimizing the distance between this predicted distribution and that representing the correct class label (the value of the next pitch). An obvious choice of evaluation measure in this case would be the information theoretic quantity which calculates this distance: relative entropy. Here we use a measure derived from it known as cross-entropy (H_c), in order to compare our results with previous work [22]. This gives us the mean divergence between the entropy calculated from the predicted distribution and that of the correct prediction label (and can be interpreted as the distance between these two distributions) for every sample in some given data. It can be computed over all the events belonging to different sequences in the test data \mathcal{D}_{test} , as

$$H_c(P_{mod}, \mathcal{D}_{test}) = \frac{-\sum_{s \in \mathcal{D}_{test}} \sum_{t=1}^{T_s} \log_2 P_{mod}(s^{(t)}|s^{(1:t-1)})}{\sum_{s \in \mathcal{D}_{test}} T_s} \quad (11)$$

¹ Website: <http://kern.ccarh.org/browse?l=essen>

where P_{mod} is the probability assigned by the model to the pitch of the event $s^{(t)}$ in the melody $s \in \mathcal{D}_{test}$ given its preceding context, and T_s is the length of s . Cross-entropy approaches the true entropy as the number of test samples, i.e., the denominator in (11) increases.

4.3 Methodology

The models are evaluated using 10-fold cross-validation. We use randomised folds identical to those used in previous work [4, 22] to facilitate fair comparison². A small part of the training set (5%) in each fold is extracted as the validation set for model selection over the various hyperparameters described below. This procedure is repeated independently for each of the 8 datasets in the corpus.

The RTDRBM LTMs were learned (offline) up to a maximum of 250 epochs using mini-batch gradient descent on the training set, and that with the best validation set score was chosen for evaluation on the test set. A grid search was carried out to determine the best set of hyperparameters for each model. These constitute the learning rate η , the L_1 and L_2 regularization (λ_1 and λ_2 respectively) and the number of hidden units n_{hid} . For each of the models, η was varied as $\{0.01, 0.05\}$, and n_{hid} as $\{10, 25, 50, 100, 200\}$. Both L_1 and L_2 decay were set to identical values $\lambda_1 = \lambda_2 = \lambda$ which was either on $(\lambda = 0.0001)$ or off $(\lambda = 0.0000)$. Learning rate was made to decay according to the schedule $\eta_t = \eta_{init}/(1 + t/\tau)$, where $\tau = 50$.

The RTDRBM LTM+s and STMs were learned (online) using stochastic gradient descent, where model parameters were updated after each time-step during prediction on the test set, with the only distinction between the two being that the parameters of the former are initialized to those of the best LTM learned offline on the dataset. As explained in Section 2.1, the LTM+ is continuously updated across melodies, while the STM is re-initialized after each melody in the test set. Since each of the STMs is expected to learn a smaller number of patterns than its corresponding LTM, we decided to extend the model selection with much smaller models as well ($n_{hid} \in \{2, 5, 10, 20, 100, 200\}$), with the remaining hyperparameters kept the same, and a constant learning rate i.e., $\eta_t = \eta_{init} = 0.01$.

The combination bias parameter b for computing the entropy-based weights w_m was varied as $b = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 16, 32\}$, as in [21]. This range was used for both combination rules, following the example of [21].

4.4 Results & Discussion

Table 3 shows the predictive performance of various LTM-STM combination rules evaluated here together with the corresponding combination bias value used, averaged across all 8 datasets. The bottom row of this table corresponds to the performance of the purely n -gram based melody model in [22], which we compare the models evaluated here with.

² Information about the training/test split in the 10 folds was obtained from the authors of [22]

Model	LTM	STM	Mix.	b_m	Prod.	b_p
(a)	2.712	3.053	2.480	3	2.496	1
(b)	2.712	3.046	2.421	4	2.487	1
(c)	2.712	3.363	2.674	5	2.703	1
(d)	2.756	3.053	2.574	2	2.563	1
(e)	2.756	3.046	2.540	2	2.581	1
(f)	2.756	3.363	2.749	5	2.773	1
n -gram	2.614	3.147	2.479	2	N/A	N/A

Table 3: Predictive performance of various model combinations listed in Table 1, in comparison with a purely n -gram based melody model (bottom row). Each row of the table contains the prediction cross-entropies of the constituent LTM (or LTM+), STM, and the combination of these two using the Mean and Product rules together with the respective biases. A lower value of cross-entropy reflects more accurate predictions.

In each case, the RTDRBM LTM has 100 hidden units (found to be the best in the model selection procedure). Despite the extended grid search for the STMs, it was found that the optimal number of hidden units was 100 in that case as well.

The first thing to note is that combining the models (using either of the two combination rules) results in an improvement in predictive performance over each of the constituent models. Furthermore, the Mean combination rule results in slightly better prediction cross-entropies than Product rule. This can be explained by considering the basic properties of the two rules, as concluded by a previous study comparing them [29]. The Mean combination rule is useful in case of identical or very highly correlated feature spaces (which holds true in the present case) in which classifiers make independent errors. Furthermore, this rule is generally more fault tolerant in the case of poor posterior probability estimates (which is indeed the case here with the STM being learned afresh at the start of each melody), whereas the Product rule emphasizes the points of agreement between the two models and is apt where classifiers make small estimation errors. The best combined model (RTDRBM LTM; n -gram (C^*I) STM) performs slightly better than the best purely n -gram based melody model in [22]. In the case of both the Mean and Product rules, it was found that smaller values of the combination bias parameter were preferred over larger ones, with a value of 1 being consistently optimal in the case of the latter.

Another observation is regarding the LTM and LTM+, where the latter performs slightly worse when compared to the former. This contrasts what has been previously observed when using n -gram models, where there was an improvement from the LTM to the LTM+ [22]. One possible reason for this could be the absence of any new sequential regularities in the test data to update the already optimized LTM with, since both the training and test sequences have been sampled from the same data distribution. Alternatively, the gradient-based optimization procedure employed here for online learning (stochastic

gradient-descent) might not be the ideal choice for updating the model quickly enough to facilitate an improvement in the predictions. The latter reason could also explain the relatively poor performance of the RTDRBM STMs when compared to the STMs based on n -grams. This requires further investigation.

5. CONCLUSIONS & FUTURE WORK

This paper presented a study on models for melody prediction with a long-term and a short-term component (LTM and STM respectively). While all the LTM models explored here are based on the Recurrent Temporal Discriminative RBM (RTDRBM), the STMs are based both on the RTDRBM and n -gram models. It was found that, while the RTDRBMs are indeed a suitable choice when learned offline as LTMs [3], they fail to achieve a predictive performance as good as that of the n -gram models considered here in an online setting (as in the case of the LTM+ and the STM). The best model in the present work is a combination of an RTDRBM LTM and an n -gram STM which performs better than the state-of-the-art model based purely on n -grams. Among the two combination rules - Mean and Product - it was found that the former rule works better with the models and data used here.

One issue that remains unresolved in the present work, and requires investigation in the future, is the lack of improvement in predictions during online learning in the RTDRBM LTM. Another extension to the models employed here is to incorporate additional melodic features as inputs, as detailed in *Multiple Viewpoints for Music Prediction* [6], and to examine how this would improve or worsen the predictive performance over the existing models. And finally, previous work with LTMs and STMs based purely on n -gram models has found the predictions made by these models to reflect the musical expectations of human subjects. This is also relevant to the models explored here, and is of interest in the future.

6. ACKNOWLEDGEMENTS

Srikanth Cherla and Son N. Tran are supported by Ph.D. studentships from City University London. The authors would like to thank Marcus Pearce (Queen Mary University of London) and Senanayak Sesh Kumar Karri (INRIA Rocquencourt) for their advice and helpful discussions related to parts of this paper.

7. REFERENCES

- [1] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Modeling Temporal Dependencies in High-Dimensional Sequences: Application to Polyphonic Music Generation and Transcription. In *International Conference on Machine Learning*, pages 1159–1166, 2012.
- [2] Stanley Chen and Joshua Goodman. An empirical study of smoothing techniques for language modeling. *Computer Speech & Language*, 13(4):359–393, 1999.
- [3] Srikanth Cherla, Son Tran, Artur d’Avila Garcez, and Tillman Weyde. Discriminative Learning and Inference in the Recurrent Temporal RBM for Melody Modelling. In *International Joint Conference on Neural Networks*, 2015.
- [4] Srikanth Cherla, Tillman Weyde, Artur d’Avila Garcez, and Marcus Pearce. A Distributed Model for Multiple-Viewpoint Melodic Prediction. In *International Society for Music Information Retrieval Conference*, pages 15–20, 2013.
- [5] Joel E Cohen. Information Theory and Music. *Behavioral Science*, 7(2):137–163, 1962.
- [6] Darrell Conklin and Ian Witten. Multiple Viewpoint Systems for Music Prediction. *Journal of New Music Research*, 24(1):51–73, 1995.
- [7] Greg Cox. On the Relationship Between Entropy and Meaning in Music: An Exploration with Recurrent Neural Networks. In *Annual Conference of the Cognitive Science Society*, pages 429–434, 2010.
- [8] Michael Cuthbert and Christopher Ariza. music21: A Toolkit for Computer-aided Musicology and Symbolic Music Data. In *International Society for Music Information Retrieval Conference*, pages 637–642, 2010.
- [9] Tuomas Eerola, Petri Toivianen, and Carol Krumhansl. Real-Time Prediction of Melodies: Continuous Predictability Judgements and Dynamic Models. In *International Conference on Music Perception and Cognition*, pages 473–476, 2002.
- [10] David Huron. *Sweet Anticipation: Music and the Psychology of Expectation*. MIT Press, 2006.
- [11] Timothy Justus and Jamshed Bharucha. Modularity in Musical Processing: The Automaticity of Harmonic Priming. *Journal of Experimental Psychology: Human Perception and Performance*, 27(4):1000–1011, 2001.
- [12] Roland Kuhn and Renato De Mori. A Cache-based Natural Language Model for Speech Recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(6):570–583, 1990.
- [13] Hugo Larochelle and Yoshua Bengio. Classification using Discriminative Restricted Boltzmann Machines. In *International Conference on Machine Learning*, pages 536–543, 2008.
- [14] Christopher Manning and Hinrich Schütze. *Foundations of Statistical Natural Language Processing*. MIT Press, 1999.
- [15] Leonard Meyer. *Emotion and Meaning in Music*. University of Chicago Press, 1956.
- [16] Leonard Meyer. Meaning in Music and Information Theory. *Journal of Aesthetics and Art Criticism*, pages 412–424, 1957.

- [17] Alistair Moffat. Implementing the PPM data compression scheme. *Communications, IEEE Transactions on*, 38(11):1917–1921, 1990.
- [18] Alistair Moffat, Radford Neal, and Ian Witten. Arithmetic Coding Revisited. *ACM Transactions on Information Systems (TOIS)*, 16(3):256–294, 1998.
- [19] Eugene Narmour. *The Analysis and Cognition of Melodic Complexity: The Implication-Realization Model*. University of Chicago Press, 1992.
- [20] Diana Omigie, Marcus Pearce, Victoria Williamson, and Lauren Stewart. Electrophysiological Correlates of Melodic Processing in Congenital Amusia. *Neuropsychologia*, 2013.
- [21] Marcus Pearce. *The Construction and Evaluation of Statistical Models of Melodic Structure in Music Perception and Composition*. PhD thesis, City University London, 2005.
- [22] Marcus Pearce and Geraint Wiggins. Improved Methods for Statistical Modelling of Monophonic Music. *Journal of New Music Research*, 33(4):367–385, 2004.
- [23] Martin Rohrmeier and Stefan Koelsch. Predictive Information Processing in Music Cognition. A Critical Review. *International Journal of Psychophysiology*, 83(2):164–175, 2012.
- [24] Jenny Saffran, Elizabeth Johnson, Richard Aslin, and Elissa Newport. Statistical Learning of Tone Sequences by Human Infants and Adults. *Cognition*, 70(1):27–52, 1999.
- [25] Helmut Schaffrath and David Huron. The Essen Folk-song Collection in the Humdrum Kern Format. 1995.
- [26] Siddharth Sigtia, Emmanouil Benetos, Nicolas Boulanger-Lewandowski, Tillman Weyde, Artur d’Avila Garcez, and Simon Dixon. A Hybrid Recurrent Neural Network For Music Transcription. In *International Conference on Acoustics Speech and Signal Processing*, 2015.
- [27] Paul Smolensky. Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 1. chapter Information Processing in Dynamical Systems: Foundations of Harmony Theory, pages 194–281. MIT Press, 1986.
- [28] Ilya Sutskever, Geoffrey Hinton, and Graham Taylor. The Recurrent Temporal Restricted Boltzmann Machine. In *Advances in Neural Information Processing Systems*, pages 1601–1608, 2009.
- [29] David Tax, Martijn Van Breukelen, Robert Duin, and Josef Kittler. Combining Multiple Classifiers by Averaging or by Multiplying? *Pattern Recognition*, 33(9):1475–1485, 2000.
- [30] Paul Werbos. Backpropagation Through Time: What It Does and How to Do It. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.
- [31] Raymond Whorley. *The Construction and Evaluation of Statistical Models of Melody and Harmony*. PhD thesis, Goldsmiths, University of London, 2013.

EFFICIENT MELODIC QUERY BASED AUDIO SEARCH FOR HINDUSTANI VOCAL COMPOSITIONS

Kaustuv Kanti Ganguli¹ Abhinav Rastogi² Vedhas Pandit¹
Prithvi Kantan¹ Preeti Rao¹

¹ Department of Electrical Engineering, Indian Institute of Technology Bombay

² Electrical Engineering, Stanford University

kaustuvkanti@ee.iitb.ac.in

ABSTRACT

Time-series pattern matching methods that incorporate time warping have recently been used with varying degrees of success on tasks of search and discovery of melodic phrases from audio for Indian classical vocal music. While these methods perform effectively due to the minimal assumptions they place on the nature of the sampled pitch temporal trajectories, their practical applicability to retrieval tasks on real-world databases is seriously limited by their prohibitively large computational complexity. While dimensionality reduction of the time-series to discrete symbol strings is a standard approach that can exploit computational gains from the data compression as well as the availability of efficient string matching algorithms, the compressed representation of the pitch time series itself is not well understood given the pervasiveness of pitch inflections in the melodic shape of the raga phrases. We propose methods that are informed by domain knowledge to design the representation and to optimize parameter settings for the subsequent string matching algorithm. The methods are evaluated in the context of an audio query based search for Hindustani vocal compositions in audio recordings via the mukhda (refrain of the song). We present results that demonstrate performance close to that achieved by time-series matching but at orders of magnitude reduction in complexity.

1. INTRODUCTION

A bandish, or composition in the North Indian classical vocal genre of khayal, is characterised by its mukhda, its almost cyclically repeated refrain. The singer elaborates within the raga framework in each rhythmic cycle before returning to the main phrase of the bandish (i.e. its mukhda). The automatic detection of this repetitive phrase, or motif, from the audio signal would contribute

to important metadata concerning the identity of the bandish. The mukhda is recognised by the lyrics, location in the cycle and its melodic shape. While these are in order of decreasing ease in terms of manual segmentation of the mukhda, the melodic shape characterized by a pitch contour segment is most amenable to pattern matching methods. The challenge here arises from the improvisatory nature of the genre where the raga grammar allows for considerable variation in the melodic shape of any prescribed phrase. Previous work has shown that the variability in the mukhda across the concert, similar to that of other raga-characteristic phrases in a performance, can be characterized as globally constrained non-linear time-warping where the constraint appears to depend on certain characteristics of the underlying melodic shape [16, 17, 21]. A dynamic time-warping (DTW) distance measure was used on the time-series segments to model melodic similarity under local and global constraints that were learned from a raga-specific corpus [17]. More recent work has also validated the DTW based similarity measure in the context of melodic motif discovery but the high computational costs associated with time-series search limited its applicability [3, 9, 14]. Given that DTW based local matching, with relatively minimal assumptions, on the pitch time-series derived from the audio is largely successful in modeling the relevant melodic variations, we focus on targeting similar performance with greatly reduced complexity. Computationally efficient methods to search and localize occurrences of the mukhda in a concert, given an isolated audio query phrase, have the following potential real-world applications: (i) automatic segmentation of all occurrences of the mukhda provided one manually identified instance, with a goal to reduce manual effort in the rich transcription of concert audio recordings, and (ii) retrieving a specific bandish from a database of concert recordings by querying by its mukhda provided either by an audio fragment or by user singing.

The acoustic correlate of the melodic shape of a phrase is its pitch contour represented computationally by the detected pitch of the singing voice at close uniformly spaced intervals. Considering the concert recording context where an instrumental ensemble accompanies the vocalist, the pitch detection is achieved by a singing voice detection algorithm coupled with predominant F0 extraction at uniformly closely spaced intervals throughout the concert. The



© Kaustuv Kanti Ganguli, Abhinav Rastogi, Vedhas Pandit, Prithvi Kantan, Preeti Rao. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Kaustuv Kanti Ganguli, Abhinav Rastogi, Vedhas Pandit, Prithvi Kantan, Preeti Rao. "EFFICIENT MELODIC QUERY BASED AUDIO SEARCH FOR HINDUSTANI VOCAL COMPOSITIONS", 16th International Society for Music Information Retrieval Conference, 2015.

pitch contour can be treated as a one-dimensional time-series which can be searched for the occurrence of a specific pattern as defined by the query (another time-series segment). We note that the dimensionality of the time-series is typically very high due to the required dense sampling of the pitch contour across the concert duration. It has been observed that a sampling interval on the order of 20 ms is necessary in order to preserve important pitch nuances as determined by the curve of rapidly decreasing correlation between melodically similar pitch contours with increasing sampling interval [9].

As mentioned earlier, DTW can be used in an exhaustive search across the concert of this sampled pitch time series to find the optimal cost alignment between the query and target pitch contours at every candidate location. We see therefore that any significant computational complexity reduction can only come from the reduction of dimensionality of the search space. An obvious choice is a representation of the melodic contour that uses compact musical abstractions such as a sequence of discrete pitch scale intervals (essentially, the note sequence corresponding to the melody if there was one). String-matching algorithms can then be applied that find the approximate longest common subsequence between the query and target segments of discrete symbols. Krannenburg [11] used this approach on audio recordings of folk songs to establish similarity in tunes across songs. Each detected pitch value was replaced by its MIDI symbol and the Smith-Waterman local sequence alignment algorithm was used on the resulting strings. Note however that there was no reduction in the size of the pitch time-series. If the pitch time-series is segmented into discrete notes, a far more compact string representation can be obtained by using each symbol to represent a tuple corresponding to a note value and duration. In this case, a number of melodic similarity methods based on the alignment of symbolic scores become available [1, 6, 11, 12, 27]. The effectiveness of this approach, of course, depends heavily on the correspondence between the salient features of the pitch contour and the symbol sequence. A specific challenge in the case of Hindustani vocal music is that it is characterized just as much by the precisely intoned raga notes as it is by the continuous pitch transitions and ornaments that contribute significantly to the raga identity, motivating a more careful consideration of the high-level abstraction [15, 18].

The main contributions of this work are (i) a study of the suitability of two distinct high-level abstractions for sequence representation in the context of our melodic phrase retrieval task, and (ii) using domain knowledge for the setting of various representation and search parameters of the systems. In the next section, we describe our test dataset of concerts with a review of musical and acoustic characteristics that are relevant to our task. This is followed by a presentation of our melodic phrase retrieval methods including approaches to the compact representation of the pitch time-series and discussion of the achievable reduction in computational complexity with respect to the baseline system. A description of the experiments follows. Finally the

results are discussed with a view to providing insights on the suitability of particular approaches to specific characteristics of the test data.

2. TEST DATABASE DESCRIPTION

The dataset comprises 50 commercial CD-quality concert audio recordings by 18 eminent Hindustani vocal artists. The accompaniment consists of tanpura (drone) and tabla, along with harmonium or sarangi. The concerts have been chosen from a large corpus [23] in a deliberate manner so as to achieve considerable diversity in artists, ragas and tempo. We restrict our analysis to the vilambit (slow tempo) and madhyalaya (medium tempo) sections of these concerts for the current task. Drut (fast tempo) sections are excluded because their mukhda phrases contain a considerable amount of context-dependent variation and hence melodic similarity is not as strongly preserved. Table 1 summarises our dataset where 39 concerts are of vilambit laya and the remaining 11 are madhyalaya. The average duration of a vilambit bandish is 17 minutes and contains an average of 20-25 mukhda instances that occur once each in a rhythmic cycle.

# Song	Dur (hrs)	# GT	Dur (hrs)	Ratio	# Unique	
					Raga	Artist
50	13:13	1075	1:44	13%	34	18

Table 1. Description of the test dataset.

Manual annotation of the mukhda segments with start and end boundaries was carried out by a musician and validated by a second very experienced musician. Mukhdas are most easily identified by listening for the lyrical phrase that occurs about the first beat (sam) of the rhythmic cycle as evidenced by the accompanying tabla strokes. The mukhda is labeled together with its boundaries as detected from the onsets of the lyric syllables. These annotations serve as the ground truth (GT) for the evaluation of the different systems under test which exploit only the similarity of melodic shape to that of the audio query. The query thus could be an instance extracted from the audio track, or it could be a sung or hummed likeness of the melodic phrase generated by the user.

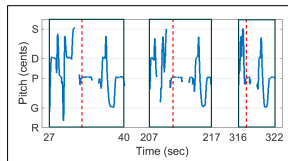


Figure 1. Pitch contour segments of distinct mukhdas. Sam of the corresponding rhythmic cycle is marked in red.

Both the cues easily available to listeners, the phones of the lyrics (as uttered by the singer) and the sam tabla strokes cannot be extracted reliably from the polyphonic audio signal. The predominant F0 extractor on the other hand is more robust and achieves the tracking of the vocalist's pitch based on dominance and continuity constraints without any explicit source separation. Our approach to mukhda detection is currently based on the computation of melodic similarity which, ideally, should encapsulate the notion of musically perceived similarity. The low-level acoustic correlate of the melody is the pitch contour, the implementation of which is presented in the next section.

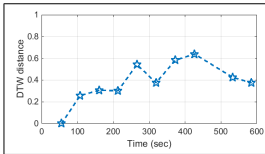


Figure 2. Normalized DTW distance between the first mukhda of the concert and subsequent mukhdas.

Figure 1 shows pitch contour segments of three mukhdas manually extracted from the beginning, middle and towards the end of the madhyalaya bandish of a concert. Also marked is the location of the sam with respect to the mukhda pitch trajectory. We note the variability in the melodic shape. Typically the tempo of the concert increases gradually over time (linked to the reduction in the rhythmic cycle duration) leading to a decrease in mukhda duration (from 13 sec to 7 sec in Figure 1). Rather than a linear compression, the melodic shape is modified by non-linear time warping [5]. Figure 2 shows a plot of DTW distance between the first mukhda of the concert and each later mukhda versus the temporal location (the corresponding sam) of the later mukhda. The distances are normalized with respect to that of the first false detection. We observe a trend of decreasing similarity with increasing time, as well as the fact that the intervals between mukhdas are not identical due to rhythmic cycle duration variability. Also, not every rhythmic cycle is marked by a mukhda. Finally, we note that the DTW distance measure is largely insensitive to the irrelevant differences, as seen from the distance values normalised with respect to the distance between the first mukhda and the nearest false detection.

3. MELODIC PHRASE RETRIEVAL SYSTEMS

In this section, we consider various approaches towards our end goal which involves searching the entire vocal pitch track extracted from the audio recording to identify pitch contour sub-segments that match the melodic shape of the query. We present the audio pre-processing required to generate the pitch time-series followed by a discussion of the different systems in terms of algorithm design and complexity.

3.1 Time series extraction from audio

The desired time-series representation is expected to capture the melody line, and hence requires accurate pitch detection of the main voice in polyphonic audio. The singing voice usually dominates over other instruments in a vocal concert performance in terms of its level and continuity over relatively large temporal extents although the accompaniment of tabla and other pitched instruments such as the drone and harmonium are present. Predominant-F0 detection is implemented by the salience based combination of two algorithms [20] which exploit the spectral properties of the voice with temporal smoothness constraints on the pitch. The pitch is detected at 20 ms intervals throughout the audio with zero pitch assigned to the detected purely instrumental regions. Next, the pitch values in Hz are converted to the cents scale by normalizing with respect the concert tonic determined by automatic tonic detection [8]. This normalization helps match a query across concerts by different artists. The final pre-processing step is to interpolate short silence regions below a threshold (80 ms which is empirically tuned in previous studies [16, 17]) indicating musically irrelevant breath pauses or unvoiced consonants by cubic spline interpolation so as to preserve the integrity of the melodic shape.

3.2 Baseline system

Our baseline method is the “subsequence DTW”, an adaptation of standard DTW to allow searching for the occurrence and alignment of a given query segment within a long sequence [13, 26]. Given a query Q of length N symbols and a much longer sequence S of length M (i.e. the song or concert sequence in our context) to be searched, a dynamic programming optimization minimizes the DTW distance to Q over all possible subsequences of S . The allowed step-size conditions are chosen to constrain the warping path to within an overall compression / expansion factor of 2. No further global constraint is applied. The candidate subsequences of the song are listed in order of increasing DTW distance to which a suitable threshold can be applied to select and localize the corresponding regions in the original audio. The time complexity of subsequence DTW is $O(MN)$ where $N(M)$ is the number of pitch samples corresponding to the query (song) duration (i.e. 50 pitch samples per second of the time series duration, given that the pitch is extracted at 20 ms intervals) [2, 13, 28]. We see that the time-series dimensions contribute directly to the complexity of the search. Our goal is to find computationally simple alternatives to DTW by moving to low dimensional string search paradigms. This requires principled approaches to converting the pitch time-series to a discrete symbol sequence, two of which are presented next.

3.3 Behavior based system

With a goal to preserve the characteristic shape of the mukhda including the pitch transitions in the mapping to the symbol sequence, we consider the approach of Tanaka [25] who proposed “behavioral symbols” to capture dis-

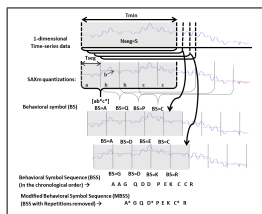


Figure 3. Construction from a pitch time series of the BS sequence (BSS) and the modified BSS.

tinct types of local temporal variation in a human motion capture system. A melodic phrase can be viewed as a sequence of musical gestures by the performer, with a behavioral symbol then potentially corresponding to a single (arbitrary movement) in pitch space. A sequence of symbols would serve as a sketch of the melodic motif. In Tanaka’s system, the symbols are purely data-dependent and evolve from the analysis itself [24, 25]. We bring in musical context constraints as presented in the algorithm description next.

The pitch time-series is segmented into fixed duration windows centered at uniformly spaced intervals so that the windows are highly overlapping as illustrated in Figure 3. The pitch contour within each window is replaced by a piecewise flat contour where each piece represents a fixed fraction of the window. While Tanaka recommends normalization of the pitch values within the window to $[0, 1]$ range in order to eliminate vertical shifts and scaling between otherwise similar shapes, we omit this step given that we are not looking for transposition or scaling invariance in the mukhda detection task. The piece-wise flat sub-segments are obtained by the median of the pitch values in the corresponding subsegment. We choose median as opposed to mean [24] as it is less sensitive to the occasional outliers in the pitch contour. We bring in further domain constraints by using the discrete scale intervals for the quantization of the piecewise sub-segments that describe a specific behavioral symbol (BS). We obtain a sequence of BS, one for each window position. Due to the high overlap between windows, repetitions are likely in consecutive symbols. These are replaced by a single BS which step brings in the needed time elasticity. Figure 3 illustrates the steps of construction of the BS sequence (BSS) and its repetition removed version (the modified BSS) from a simulated pitch time-series.

The database is pre-processed and the symbol sequence representation of each complete concert recording is stored. When a query is presented, it is converted to its symbol sequence (which currently depends on the song to be searched) and an exact sub-sequence search is implemented on the song string. The choice of the fixed parameters: window duration, hop duration and number of subsegments within a window turn out to heavily influence the representation. The window duration should depend on the time scale of the salient features (movements in

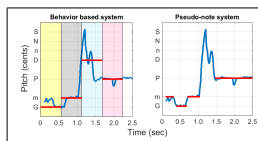


Figure 4. The two proposed systems of quantization, namely: behavior based and pseudo-note systems.

pitch space). The subsegments must be small enough to retain the melodic shape within the window. The hop of the sliding window compensates for alignment differences of the different occurrences of the template in the pitch time-series of the song. We present “parameter settings” for two configurations.

Version A: Fixed parameter setting (window = 126 samples, hop = 5 samples, # subsegments per window = 3)

Version B: Query dependent setting (window = $(0.5 * N)$ samples, hop = 5 samples, # subsegments per window = 4)

We present next an alternate approach to symbolic representation of the pitch contour.

3.4 Pseudo-note system

An approximation to staff notation can be achieved by converting the continuous time-series to a sequence of piecewise flat segments if the section pitches are chosen from the set of discrete scale intervals of the music. If the achieved representation indeed corresponds to some underlying skeleton of the melodic shape of the phrase, we could anticipate obtaining better matches across variations of the melodic phrase. We address the question of how we can bring domain knowledge into this transformation. As we see from Figure 4, the continuous pitch contours corresponding to the phrases are not directly suggestive of a specific sequence of raga notes given that raga notes are embellished considerably when realized by the vocalist. In Indian music traditions, written notation has a purely prescriptive role and achieving the transcription of a performed phrase to written notation requires raga knowledge and much experience [19]. All the same there is a similarity across the mukhda repetitions that we wish to capture in our representation.

We consider a simple representation of the melodic shape that features only the relatively stable regions of the continuous pitch contours that lie within a musically valid interval of a scale (raga) notes. The scale notes are detected from the prominent peaks of the long-term pitch histogram across the concert and the musically valid interval is chosen to be within 35 cents [17]. This step leaves fragments of the time-series that coincide with the scale notes while omitting the remaining pitch transition regions. Next, a lower threshold duration of 80 ms is applied to the fragments to discard fragments that are considered too short to be perceptually meaningful as held notes [16]. This leaves a string of fragments each labeled by a svara (raga note as shown in Figure 4 (right)). Fragments with the same note value that are separated by gaps less than 80 ms are

merged. The resulting symbol sequence thus comprises the scale notes occurring in the correct temporal order but without explicit durational information. The database is pre-processed and the symbol sequence representation of each complete concert recording is stored. When a query is presented, it is converted to its symbol sequence and an approximate sub-sequence search is implemented on the concert string based on an efficient string matching algorithm with parameter settings that are informed by domain knowledge as described next.

The similarity measurement of the query sequence with candidate subsequences of the song is based on the Smith-Waterman algorithm, widely used in bioinformatics but also applied recently to melodic note sequences [11, 22]. It performs the local alignment of two sequences to find optimal alignments using two devices. A symbol of one sequence can be aligned to a symbol of the other sequence or it can be aligned to a gap. Each of these operations has a cost that is designed as follows.

Substitution score: In its standard form, the Smith-Waterman algorithm uses a fixed positive cost for an exact match and a fixed negative score for symbol mismatch. In the context of musical pitch intervals, we would rather penalize small differences less than large differences. We present alternate substitution score functions that incorporate this.

Gap Function: This function deducts a penalty from the similarity score in the event of insertion or deletion of symbols during the alignment procedure. The default gap penalty is linear, meaning that the penalty is linearly proportional to the number of symbols that comprise the gap. Another possibility, that is more meaningful for the melody context, is the affine gap function where the gap opening cost is high compared to the cost incurred by adding each successive symbol to the gap [7]. This is achieved by a form given by $mx + c$ where x is the length of the gap and m, c are constants. Intuitively, increasing c will penalize gap openings to a greater extent, while increasing m will have a similar effect with regard to gap extension. We present different designs for the relative costs motivated by the musical context.

With variations in each of the above two controls of the Smith-Waterman algorithm, we obtain the following three distinct versions of the pseudo-note system.

Version A: This setting is similar to the default Smith-Waterman setting, with a distance-independent similarity function that assesses a score of +3 for symbol match and -1 for a substitution. Gap function is linear, with penalty equal to symbol length of gap.

Version B: Substitution score that takes pitch difference into account, i.e. Score of +3 for a match, 0 for symbols differing by upto 2 semitones, -1 for substitution, and an affine gap penalty with parameters $m = 0.8, c = 1$.

Version C: Query dependent settings where we use the settings of B as default with the following changes for particularly fast varying and slowly varying query melodic shapes as determined by a heuristic measure of ratio of squared number of symbols to query duration. We have the fol-

lowing parameter settings. (i) fast varying: Substitution score of +1 to symbols differing by upto 2 semitones. Gap penalty is affine with parameters $m = 1, c = 0.5$, and (ii) slowly varying: Similarity score of -0.5 to symbols differing by upto 3 semitones. Gap penalty is affine with parameters $m = 0.5, c = 1.5$.

Finally, the Smith-Waterman algorithm has a time complexity given by $O(MN^2)$ where N is the query length in symbols and M is the song length [22]. By constraining the allowed gap length to be no longer than that of the query itself (N), justified by the musical context, we achieve a complexity reduction to $O(MN)$.

4. EXPERIMENTS AND EVALUATION

We present experiments that allow us to compare the performance of the different systems on the task at hand, namely correctly detecting occurrences of the mukhda in the audio concert given an audio query corresponding to the melodic shape of the mukhda phrase. The queries are drawn from a set of 5 mukhdas extracted from the early part (first few cycles) of the bandish. The early mukhda repetitions tend to be of the canonical form and hence correspond well with an isolated query that a musician might generate to describe the bandish. For the investigation of a given method, we process the database to convert each concert audio to the pitch time series and then to the corresponding string representation. Next, the query is converted to the string representation and the search is executed. The detections with time-stamps are listed in order of decreasing similarity with the query as determined by the corresponding search distance measure. A detection is considered a true positive if the time series of the detection spans at least 50% of that of one of the ground-truth labeled mukhdas in the song. An ROC (precision vs recall) is obtained for each query by sweeping a threshold across the obtained distances. The ROC for a song is derived by the vertical averaging (i.e. recall fixed and precision averaged) of the ROCs of the 5 distinct queries [4]. The performance for each song is summarized by the following two measures: precision at 50% recall and the equal error rate (EER) (point on the ROC at which false acceptance rate matches false rejection rate). We further present performance of the best performing pseudo-note system on song retrieval in terms of the mean reciprocal rank (MRR) [10] on the dataset of 50 concerts as follows. We use the set of the first occurring labeled mukhda of each song to form a test set of 50 queries. Next for each test query, every song is searched to obtain a rank-ordered list of songs whose first 5 detections yield the lowest averaged distance measure to the query.

5. RESULTS AND DISCUSSION

Table 2 compares the performances of the various systems on the task of mukhda detection in terms of the average EER and average precision at a selected recall across the 50 songs where each song is queried using each of the first five mukhdas. We also report the computational complexity

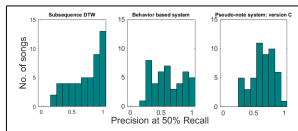


Figure 5. Histogram of the measure ‘Precision at 50% Recall’ across the baseline and proposed methods.

reduction factor over that of the baseline method (given by the square of the dimension reduction factor). To obtain more insight into song dependence, if any, we show the distribution of the precision values for the 50 songs set in the bar graphs of Figure 5, one system for each category, represented by the best performing one.

Method (version)	Mean EER	Prc at 50% Rec		Reduc.	
		Mean	Std.		
Subseq DTW	—	0.33	0.73	0.18	1
Behavior based system	(A)	0.47	0.56	0.26	100
	(B)	0.41	0.61	0.25	
Pseudo-note system	(A)	0.47	0.61	0.19	2500
	(B)	0.42	0.64	0.19	
	(C)	0.41	0.65	0.18	

Table 2. Comparison of the two performance measures and computational complexity reduction factor across the baseline and proposed methods.

From Table 2, we observe that the baseline system represented by subsequence DTW on the pitch time-series performs the best while the pseudo-note methods achieve the best computation time via a reduction proportional to the square of the reported dimension reduction factor (i.e. 50). We will first comment on the relative strengths of these two systems, and later discuss the behavior based system. We observe an improvement in performance of the pseudo-note system with the introduction of domain knowledge and query dependent parameter settings for the subsequence search algorithm. From Figure 5, we see that the subsequence DTW has a right-skewed distribution indicating a high retrieval accuracy for a large number of songs. However we note the presence of low performing songs too which actually do better with the pseudo-note system. Closer examination of these songs revealed that these belonged to ragas characterized by heavily ornamented phrases. In the course of improvisation, the mukhda was prefaced by rapidly oscillating pitch due to the preceding context. This led to increased DTW distance between the query and mukhda instances. The oscillating prelude was absent in the pseudo-note representation altogether leading to a better match.

The behavior based system was targeted towards capturing salient features of the melodic shape of the phrase in a

symbolic representation. The salient features should ideally include steady regions as well as specific movements in pitch space that contribute to the overall melodic shape. As such, it was expected to perform better than the pseudo-note method which retains relatively sparse information as seen from a comparison of the two representations for an example phrase in Figure 4. However, the selection of the duration parameters required for the time-series conversion turned out to be crucial to the accuracy of the system. Shortening the window hop interval contributed to reduced sensitivity to time alignment differences but at the cost of reduced compression and therefore much higher time complexity. Further, the data dependence of symbol assignment requires the query to be re-encoded for every song to be searched, and further if query dependent window length is chosen, the song must be re-encoded according to the query. Future work should target obtaining a fixed dictionary of symbols to pitch movement mappings by learning on a large representative database of concerts.

Top ‘M’ hits	Correct songs	Accuracy
1	41 / 50	0.82
2	45 / 50	0.90
3	48 / 50	0.96

Table 3. Results of the song retrieval experiment.

Finally, we note the song retrieval performance of the pseudo-note version C in Table 3. The mean reciprocal rank (MRR) is 0.89. The top-3 ranks return 48 of the 50 songs correctly. The badly ranked songs were found to be narrowly superseded by other songs from the same raga that happened to have phrases similar to the mukhda of the true song. This suggests the potential of the method in the retrieval of “similar” songs where the commonality of raga is known to be an important factor.

In summary, the melodic phrase is a central component for audio based search for Hindustani music. Given the improvisational nature of the genre as well as the lack of standard symbolic “notation”, time-series based matching of pitch contours provides a reasonable performance at the cost of complexity. The conversion to a relatively sparse representation by retaining only flat regions of the pitch contour and introducing domain driven cost functions in the string search is shown to lead to a slight reduction in retrieval accuracy while reducing complexity significantly. The inclusion of further cues such as the lyrics and rhythmic cycle markers to mukhda detection is expected to improve precision and is the subject of future research.

6. ACKNOWLEDGEMENT

This work received partial funding from the European Research Council under the European Union’s Seventh Framework Programme (FP7/2007-2013)/ERC grant agreement 267583 (CompMusic).

7. REFERENCES

- [1] N. Adams, M. Bartsch, J. Shifrin, and G. Wakefield. Time-series alignment for Music Information Retrieval. In *Proc. of Int. Soc. for Music Information Retrieval (ISMIR)*, pages 303–310, 2004.
- [2] A. Chan. An analysis of pairwise sequence alignment algorithm complexities. Technical report, Stanford University, 2004.
- [3] R. B. Dannenberg and N. Hu. Pattern discovery techniques for music audio. *Journal of New Music Research (JNMR)*, 32(2), 2002.
- [4] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 2006.
- [5] K. K. Ganguli and P. Rao. Tempo dependence of melodic shapes in Hindustani classical music. In *Proc. of Frontiers of Research on Speech and Music (FRSM)*, pages 91–95, March 2014.
- [6] C. Gomez, S. Abad-Mota, and E. Ruckhaus. An analysis of the Mongeau-Sankoff algorithm for Music Information Retrieval. In *Proc. of Int. Soc. for Music Information Retrieval (ISMIR)*, pages 109–110, 2007.
- [7] O. Gotoh. An improved algorithm for matching biological sequences. *Journal of Molecular Biology*, 162:705–708, 1982.
- [8] S. Gulati, A. Bellur, J. Salamon, H. G. Ranjani, V. Ishwar, H. A. Murthy, and X. Serra. Automatic tonic identification in Indian art music: Approaches and Evaluation. *Journal of New Music Research*, 43(1):53–71, 2014.
- [9] S. Gulati, J. Serra, V. Ishwar, and X. Serra. Mining melodic patterns in large audio collections of Indian art music. In *Proc. of Int. Conf. on Signal Image Technology & Internet Based Systems (SITIS)*, 2014.
- [10] Z. Guo, Q. Wang, G. Liu, J. Guo, and Y. Lu. A music retrieval system using melody and lyric. In *Proc. of IEEE Int. Conf. on Multimedia & Expo*, 2012.
- [11] P. V. Kranenburg. *A Computational Approach to Content-Based Retrieval of Folk Song Melodies*. PhD thesis, October 2010.
- [12] M. Mongeau and D. Sankoff. Comparison of musical sequences. *Computers and the Humanities*, 1990.
- [13] M. Muller. *Information Retrieval for Music and Motion, Chapter 4: Dynamic Time Warping*, pages 69–84.
- [14] M. Muller, N. Jiang, and P. Grosche. A robust fitness measure for capturing repetitions in music recordings with applications to audio thumbnailing. *IEEE Trans. on Audio, Speech & Language Processing*, 21(3):531–543, 2013.
- [15] D. Raja. *Hindustani Music: A Tradition in Transition*. D. K. Printworld, 2005.
- [16] P. Rao, J. C. Ross, and K. K. Ganguli. Distinguishing raga-specific intonation of phrases with audio analysis. *Ninaad*, 26-27(1):59–68, December 2013.
- [17] P. Rao, J. C. Ross, K. K. Ganguli, V. Pandit, V. Ishwar, A. Bellur, and H. A. Murthy. Classification of melodic motifs in raga music with time-series matching. *Journal of New Music Research (JNMR)*, 43(1):115–131, April 2014.
- [18] S. Rao, J. Bor, W. van der Meer, and J. Harvey. *The Raga Guide: A Survey of 74 Hindustani Ragas*. Nimbus Records with Rotterdam Conservatory of Music, 1999.
- [19] S. Rao and P. Rao. An overview of Hindustani music in the context of Computational Musicology. *Journal of New Music Research (JNMR)*, 43(1), April 2014.
- [20] V. Rao and P. Rao. Vocal melody extraction in the presence of pitched accompaniment in polyphonic music. *IEEE Trans. on Audio, Speech & Language Processing*, 18(8), 2010.
- [21] J. C. Ross, T. P. Vinutha, and P. Rao. Detecting melodic motifs from audio for Hindustani classical music. In *Proc. of Int. Soc. for Music Information Retrieval (ISMIR)*, October 2012.
- [22] T. F. Smith and M. S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147:195–197, 1981.
- [23] A. Srinivasamurthy, G. K. Koduri, S. Gulati, V. Ishwar, and X. Serra. Corpora for music information research in Indian art music. In *Proc. of Int. Computer Music Conf. / Sound and Music Computing Conf.*, September 2014.
- [24] Y. Tanaka, K. Iwamoto, and K. Uehara. Discovery of time-series motif from multi-dimensional data based on MDL principle. *Machine Learning*, 58:269–300, 2005.
- [25] Y. Tanaka and K. Uehara. Discover motifs in multi-dimensional time-series using the Principal Component Analysis and the MDL principle. In *Proc. of Int. Conf. on Machine Learning & Data Mining in Pattern Recognition*, pages 252–265, 2003.
- [26] P. Tormene, T. Giorgino, S. Quaglini, and M. Stefanelli. Matching incomplete time-series with Dynamic Time Warping: An algorithm and an application to post-stroke rehabilitation. *Artificial Intelligence in Medicine*, 45(1):11–34, 2008.
- [27] A. Uitdenbogerd and J. Zobel. Melodic matching techniques for large music databases. In *Proc. of ACM Int. Conf. on Multimedia*, pages 57–66, 1999.
- [28] A. Vahdatpour, N. Amini, and M. Sarrafzadeh. Towards unsupervised activity discovery using multi-dimensional motif detection in time-series. In *Proc. Of Int. Joint Conf. on Artificial Intelligence (IJCAI)*, 2009.

MODIFIED PERCEPTUAL LINEAR PREDICTION LIFTERED CEPSTRUM (MPLPLC) MODEL FOR POP COVER SONG RECOGNITION

Ning Chen¹ J. Stephen Downie² Haidong Xiao³
Yu Zhu¹ Jie Zhu⁴

¹ Dept. of Elec. and Comm. Eng., East China Univ. of Sci. and Tech., CHN

² Graduate School of Library and Information Science, UIUC, USA

³ Shanghai Advanced Research Institute, Chinese Academy of Sciences, Shanghai, CHN

⁴ Dept. of Electronic Engineering, Shanghai Jiao Tong University, CHN

nchen@ecust.edu.cn

ABSTRACT

Most of the features of Cover Song Identification (CSI), for example, Pitch Class Profile (PCP) related features, are based on the musical facets shared among cover versions: melody evolution and harmonic progression. In this work, the perceptual feature was studied for CSI. Our idea was to modify the Perceptual Linear Prediction (PLP) model in the field of Automatic Speech Recognition (ASR) by (a) introducing new research achievements in psychophysics, and (b) considering the difference between speech and music signals to make it consistent with human hearing and more suitable for music signal analysis. Furthermore, the obtained Linear Prediction Coefficients (LPCs) were mapped to LPC cepstrum coefficients, on which liftering was applied, to boost the timbre invariance of the resultant feature: Modified Perceptual Linear Prediction Liftered Cepstrum (MPLPLC). Experimental results showed that both LPC cepstrum coefficients mapping and cepstrum liftering were crucial in ensuring the identification power of the MPLPLC feature. The MPLPLC feature outperformed state-of-the-art features in the context of CSI and in resisting instrumental accompaniment variation. This study verifies that the mature techniques in the ASR or Computational Auditory Scene Analysis (CASA) fields may be modified and included to enhance the performance of the Music Information Retrieval (MIR) scheme.

1. INTRODUCTION

Cover Song Identification (CSI) refers to the process of identifying an alternative version, performance, rendition, or recording of a previously recorded musical piece [26]. It has a wide range of applications, such as music collection search and organization, music rights management and li-

censes, and music creation aids. Inspired by the actual application requirements and researchers' growing interest in identifying near-duplicated versions, CSI has become a dynamic area of study in the Music Information Retrieval (MIR) community over the past decades. As a result, for the first time in 2006, the CSI task was included by the Music Information Retrieval Evaluation eXchange (MIREX), an international community-based framework for the formal evaluation of MIR systems and algorithms [6].

Since there are many different formats of cover version, such as remastering, instrumental, mashup, live performance, acoustic, demo, remix, quotation, medley, and standard, the cover version may differ from the original in timbre, tempo, timing, structure, key, harmonization, lyrics and language, and noise [24]. What remain almost invariable among cover versions are melody evolution and harmonic progression, which form the basis of most existing CSI feature extraction algorithms. Among these features, the Pitch Class Profile (PCP) (or chroma) [9] and related descriptors [3, 7, 19, 25, 26, 31, 33]—which can represent harmonic progression directly—are robust to noise (e.g. ambient noise or percussive sounds) and independent of timbre, played instruments, loudness, and dynamics, have become the most widely-used features for CSI. In [7], the beat-synchronous chroma for two tracks were cross-correlated, from the results of which the sharp peaks indicating good local alignment were looked for to determine the distance between them. This CSI scheme performed the best in the audio CSI task contest of the 2006 MIREX. The Harmonic Pitch Class Profile (HPCP) feature proposed in [12] shared the common properties of PCP, but since it was only based on the peaks of the spectrum within a certain frequency band, it reduced the influence of noisy spectral components. It also took the presence of harmonic frequencies into account and was tuning independent. The CSI scheme based on the HPCP and Q_{max} similarity measure [26, 27] achieved the highest identification accuracy in the audio CSI task contest of the 2009 MIREX. In [19], the lower pitch-frequency cepstral coefficients were discarded and the remaining coefficients were projected onto chroma bins to obtain the Chroma DCT-Reduced log Pitch (CRP) feature. The CRP feature achieved high degree of timbre



© Ning Chen, J. Stephen Downie, Haidong Xiao, Yu Zhu, Jie Zhu. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Ning Chen, J. Stephen Downie, Haidong Xiao, Yu Zhu, Jie Zhu. "Modified Perceptual Linear Prediction Liftered Cepstrum (MPLPLC) Model for Pop Cover Song Recognition", 16th International Society for Music Information Retrieval Conference, 2015.

invariance and, thus, outperformed conventional PCP in the context of music matching and retrieval applications.

We observed that despite the promising achievements of the CSI technique over the last decade, the available CSI schemes cannot perform as well as the human ear does. One possible reason is that the available CSI schemes pay attention solely to the musical facets (e.g. melody evolution and harmonic progression) that are shared among cover versions and do not resemble the way humans process music information at all [24]. In this paper, we propose a perceptually inspired model called the MPLPLC model to process music signals based on the Perceptual Linear Prediction (PLP) model [13] in the ASR field. In the proposed scheme, we will consider equally the various attributes of human auditory processing, the difference between speech and music signals, and the requirements of representing the musical facets shared among cover versions. First, the MPLPLC model uses the Blackman window but not the Hamming window to weight each frame to maintain the harmonic information of the music. Second, it replaces frequency warping on the bark scale with a real filter bank equally spaced on the Equivalent Rectangular Bandwidth (ERB) scale to model the time and frequency resolution of human ears. Third, it substitutes a fixed equal loudness curve for a loudness model suitable for time-varying sounds (speech or music) [11]. Fourth, the hair cell transduction model [17] takes the place of cubic-root intensity-loudness compression to replicate the characteristics of auditory nerve responses, including rectification, compression, spontaneous firing, saturation effects, and adaptation [32]. Last and most important, to make the resulted feature (MPLPLC) suited for the CSI task, the LPCs are transformed into LPC cepstrum coefficients to reduce the correlation between them and their unnecessary sensitivity, the result of which is lifted to achieve some degree of timbre invariance [1, 14].

The identification power and robustness to the variation in instrumental accompaniments of MPLPLC were tested on two different collections. The first was composed of 502 songs and 212 cover sets and the second consisted of 85 cover sets whose cover versions have been performed by the same artist with different instrumental accompaniments. We observed that MPLPLC achieved higher identification accuracy, in terms of the Mean of Average Precision (MAP), the total number of identified covers in the top five (TOP-5), the mean rank of the first identified cover (RANK), and the Mean averaged Reciprocal Rank (MaRR) [23]. It also achieved a higher degree of invariance to instrumental accompaniments than the conventional PLP feature [13] and different PCP-related features: the beat-synchronous chroma [7], the HPCP [12, 26], and the CRP [19]. Experimental results also verified that both the LPC cepstrum coefficients mapping and the cepstrum lifting are crucial in ensuring the identification power of MPLPLC.

The rest of this paper is organized as follows. The signal processing steps involved in the proposed MPLPLC model have been described in detail in Section 2. The perfor-

mances of the MPLPLC feature in the CSI task in comparison with PLP and other state-of-the-art features have been evaluated and discussed in Section 3. Conclusions and prospects on future work have been given in Section 4.

2. MPLPLC MODEL

A block diagram of the MPLPLC model is shown in Figure 1. The signal processing steps involved in this model are discussed in detail as follows.

2.1 Pre-processing

The input music signal is first converted to mono, 8 kHz and 16 bits per sample version to reduce both the computation time and memory requirements. Then, it is filtered by a preemphasis filter of the form

$$H(z) = 1 - \mu z^{-1} \quad (1)$$

where the coefficient μ is chosen between 0.95 and 0.99. The preemphasis is needed because first, it weakens the influence of low-frequency noise and strengthens the high-frequency signal; second, it reduces the dynamic range of the spectrum to make autoregressive modelling easier [4]; and third, it has been proven helpful in maintaining harmonic information in audio signals [22].

2.2 Enframing

The pre-processed signal is segmented into overlapping frames, denoted as $\{s_i | i = 1, \dots, N\}$, and each frame is windowed by the Blackman window [20] to get $\{s_{w,i} | i = 1, \dots, N\}$.

We chose the Blackman window but not the Hamming window because the Blackman window has a wider main-lobe and lower highest side-lobe than the Hamming window [28]. As described in the open course *Audio Signal Processing for Music Applications*¹, this characteristic of the Blackman window helps to maintain and smooth the peaks in the spectrum corresponding to the harmonics in the music signal.

2.3 Equal Loudness Predicting

To compensate for the frequency-dependent transmission characteristics of the outer ear (pinna and ear canal), the tympanic membrane, and the middle ear (ossicular bones), each windowed frame $s_{w,i}$ is filtered by an equal loudness model to simulate the transfer function from the sound field to the oval window of the cochlea [2] to get $s_{ul,i}$. In PLP, a fixed equal-loudness curve is combined [13]. However, since a music signal is time-varying and has both short-term loudness (the loudness of a specific note) and long-term loudness (the loudness of a musical phase) [18], the fixed loudness curve is not suited to it. So, Glasberg and Moore's [11] loudness model, which can be applied directly to the sound and works for time-varying sounds, is applied to the MPLPLC model.

¹ <https://class.coursera.org/audio-001/lecture/53>

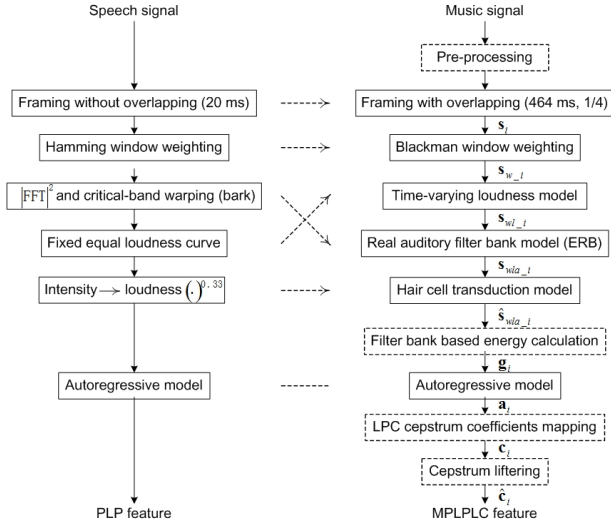


Figure 1. The comparison between the PLP model (left) and MPLPLC model (right).

2.4 Auditory Filter Bank Modeling

To obtain the auditory spectrum, PLP does a critical-band integration after a Fourier Transform (FT) [13]. The problem is that frequency bin in FT is linear, so it has a constant spectral resolution, while the human ear has high spectral resolution at low frequency and low spectral resolution at high frequency. Therefore, in the proposed scheme, a real filter-bank composed of N_f channels equidistantly spaced on the ERB [10] scale was applied to imitate the frequency resolution of human hearing. The bandwidths of the channels in the filter bank are proportional to the center frequencies (see Figure 2). The real filter bank can obtain a good spectral resolution at low frequencies and a good temporal resolution at high frequencies (like the human ear) [15]. Another advantage of the filter bank approach is that each bandpass channel is treated essentially independently, i.e., there are no global spectral constraints on the filter bank outputs [14]. In this specific case, a Hanning window on the frequency side was chosen² and the experimental results showed that the type of filter has little influence on the obtained cepstral feature. The output of the j -th channel in the filter bank for the input $s_{wla,i}$ is denoted as $s_{wla,i}^{(j)}$.

2.5 Hair Cell Transduction

In PLP [13], the cubic-root amplitude compression is combined to approximate the power law of hearing and simulate the nonlinear relation between the intensity of the

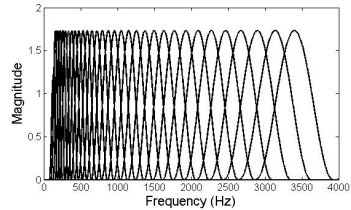


Figure 2. Frequency responses of the filters in the auditory filter bank, with center frequencies equally spaced between 131 Hz and 3400 Hz on the ERB-rate scale.

sound and its perceived loudness [29]. Meddis's hair cell transduction model [17] is incorporated in the MPLPLC model to simulate the rectification, compression, spontaneous firing, saturation effects, and adaptation characteristics of auditory nerve responses [32]. This operation also helps to reduce the spectral amplitude variation of the auditory spectrum, which makes it possible to do the all-pole modeling by a relative low model order [13]. The hair cell transduced version of $s_{wla,i}^{(j)}$ is denoted as $\hat{s}_{wla,i}^{(j)}$.

2.6 Filter Bank Based Energy Calculation

To represent the energy distribution of the music signal on each channel, the energy of the j -th channel for the i -th frame, denoted as $g_i(j)$, is calculated as follows:

² <http://tftat.sourceforge.net/doc/filterbank/erbfilters.php>

$$g_i(j) = \log \sum_{n=1}^{L_w} \left(\hat{s}_{wla,i}^{(j)}(n) \right)^2 \quad (2)$$

Here, $\hat{s}_{wla,i}^{(j)}(n), n = 1, \dots, L_w$ is the element of the vector $\hat{\mathbf{s}}_{wla,i}^{(j)}$. Then, the filter bank based energy of the i -th frame is $\mathbf{g}_i = [g_i(1), \dots, g_i(N_f)]$.

2.7 Autoregressive Modeling

To represent the spectral envelope of the filter bank based energy in a compressed form, the filter bank based energy $\mathbf{g}_i, i = 1, \dots, N$ are modelled by a p th-order all pole spectrum $\sigma/A_i(z)$, where σ is constant and $A_i(z) = 1 + a_{i1}z^{-1} + \dots + a_{ip}z^{-p}$, using the autocorrelation method [16]. Then, the LPCs of the i th frame are denoted as $\mathbf{a}_i = [a_i(1), \dots, a_i(p)]$.

2.8 LPC Cepstrum Coefficients Mapping

To reduce the correlation between them [5], the LPCs \mathbf{a}_i are further transformed into (real) LPC cepstrum coefficients, denoted as $\mathbf{c}_i = [c_i(1), \dots, c_i(p)]$, with the following recursion formula [14]:

$$c_i(n) = -a_i(n) - \frac{1}{n} \sum_{k=1}^{n-1} (n-k)a_i(k)c_i(n-k) \quad (3)$$

Figure 3(a) and 3(b) show the comparison between the spectrum of filter bank based energy and its LPC smoothing result, and that between the spectrum of filter bank based energy and its cepstrum smoothing result, respectively. It can be seen that first, both the LPC and the corresponding LPC cepstrum can represent the rough change trend of the spectral envelop of the filter bank based energy, and second, the LPC smoothing does not follow the slow variations of the filter bank based energy as well as LPC cepstrum smoothing does. This means that the LPC cepstrum mapping helps to reduce the unnecessary sensitivity that exists in LPC smoothing results.

2.9 Cepstrum Liftering

It has been proven that the variability of low frequency terms is primarily due to variation in transmission, speaker characteristics, and vocal efforts of the human voice [14]. As for the music, the lower frequency is closely related to the aspect of timbre [19, 21, 30]. So, to boost the degree of timbre invariance of the proposed feature, the liftering window proposed in [14] [see Eq.(4)] is applied to the LPCs first; then, the lower q elements of the result are truncated to get the liftered LPCs denoted as $\hat{\mathbf{c}}_i = \{\hat{c}_i(1), \dots, \hat{c}_i(p-q)\}$.

$$W_L(n) = \begin{cases} 1 + \frac{p}{2} \sin\left(\frac{\pi n}{p}\right), & n = 1, 2, \dots, p \\ 0, & \text{otherwise} \end{cases} \quad (4)$$

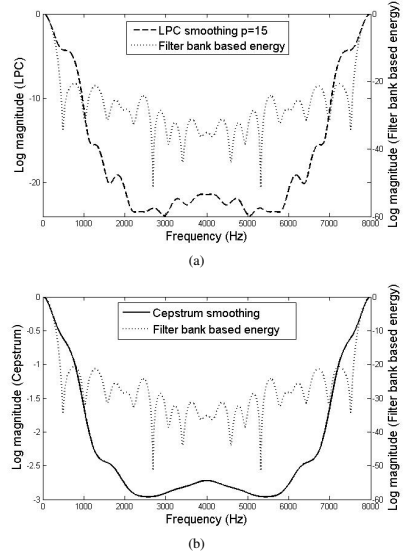


Figure 3. Comparison of spectral smoothing methods.

3. EVALUATION

3.1 Evaluation Preparation

To test the effectiveness of the MPLPLC feature in the pop CSI task, the enhanced Q_{max} method [27] (denoted as Q_{max} in this paper) was used to measure the distance between the MPLPLC time series of two pieces of music. The parameters chosen to calculate cross recurrence plots [34] were embedding dimension $m = 15$, time delay (in units) $\tau = 2$ and the maximum percentage of neighbours $\kappa = 0.1$. Furthermore, the parameters used to compute a cumulative matrix Q [26] are the penalty for a disruption onset $\gamma_o = 5$ and the penalty for a disruption extension $\gamma_e = 0.5$.

Two music collections were used. The first one (denoted as Collection_1) comprised 502 pop songs of various styles and genres and 212 cover sets. The average number of covers in each cover set is 2.4, and the distribution of the cover set cardinality has been presented in Figure 4. Western songs and Chinese songs occupy one half of this collection. The second one (denoted as Collection_2) is independent of Collection_1 and comprised 175 songs and 85 cover sets. The cover versions of each cover set in Collection_2 were pop songs performed by the same artist but with different instrumental accompaniments. The materials were obtained from a personal music collection. The identification accuracy and robustness against variation in instrumental accompaniments of the MPLPLC was tested on Collection_1 and Collection_2, in comparison with the

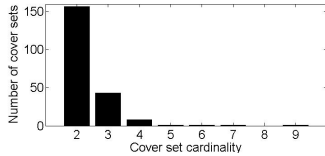


Figure 4. Distribution of the cover set cardinality.

PLP feature [13], CRP feature [19]³, Ellis’s cover song scheme [7]⁴, and Serra’s cover song scheme [27]⁵. The parameters of the MPLPLC model have been listed in Table 1, and those of PLP, CRP, Ellis’s scheme, and Serra’s scheme are the same as those in [13], [19], [7], and [27], respectively.

Table 1. Parameter setting of MPLPLC feature

Description	Value
Preemphasis parameter μ	0.97
Frame length	464ms
Frame overlap	116ms
Minimum central frequency of auditory filter	133Hz
Maximum central frequency of auditory filter	6856Hz
Number of channels in auditory filter bank N_f	41
LPC order p	16
Number of cepstrum	16
Cepstrum truncate number q	3

3.2 Identification Accuracy

We used each of the 502 songs in Collection_1 as a query and calculated the distance [27] between each query and the remaining 501 songs based on different features. The identification accuracy, in terms of TOP-5, MAP, RANK, and MaRR, obtained from the distance matrices (see Table 2) demonstrated that MPLPLC performed better than the conventional features in the CSI task over Collection_1. One possible explanation for this result is that Collection_1 was composed of pop songs that included a singing voice, and due to the MPLPLC’s background in speech recognition, it outperformed the musical facet based features in representing the singing voice. As an example, we studied two versions of the song *Wishing We Last Forever* as performed by Teresa Teng and Faye Wong, respectively. In these two versions, the singing voice is dominant, the instrumental accompaniments are different, and the rhythm is smoothing. The version performed by Teresa Teng includes a national instrument accompaniment, which doesn’t conform to the twelve-tone equal temperament. The cross recurrence plots for these two versions based on MPLPLC, CRP [19], beat-synchronous chroma [7] and HPCP [27] have been presented in Figure 5(a)-(d), respectively. We observe that the extended pattern in Figure 5(a), which corresponds to similar sections in two versions, is much more distinct and longer

than those in Figure 5(b)-(d). This indicates that first, MPLPLC may outperform the other features in representing the singing voice characteristics, and second, the difference in harmonic information resulting from the difference in instrumental accompaniment affects the performance of PCP-based features.

Table 2. The identification accuracy comparison among MPLPLC and conventional features over Collection_1.

System	Identification accuracy			
	TOP-5	MAP	RANK	MaRR
MPLPLC + \hat{Q}_{max}	738	0.9446	3.79	0.4387
PLP [13] + \hat{Q}_{max}	386	0.4783	58.52	0.2392
CRP [19] + \hat{Q}_{max}	525	0.6719	56.48	0.3237
Ellis’s [7]	600	0.7489	28.32	0.3507
Serra’s [27]	558	0.7266	28.28	0.3507

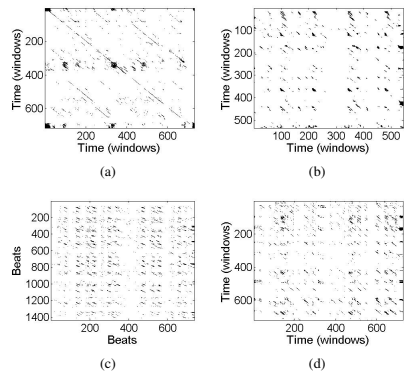


Figure 5. Cross recurrence plot for two versions of *Wishing We Last Forever* as performed by Teresa Teng and Faye Wong based on different features: (a) MPLPLC ($\hat{Q}_{max} = 464.5$), (b) CRP ($\hat{Q}_{max} = 21$), (c) Beat-synchronous chroma ($\hat{Q}_{max} = 61.5$), and (d) HPCP ($\hat{Q}_{max} = 47.5$)

3.3 Robustness against Variation in Instrumental Accompaniments

When compared with classical music, popular music can present a richer range of variation in style and instrumentation [8]. To test the robustness of MPLPLC against variation in style and instrumentation, the identification accuracy in terms of MAP achieved by MPLPLC and by the conventional features were tested and compared with Collection_2. The experimental results shown in Figure 6 indicate that the MPLPLC feature achieves a higher degree of invariance against instrumental accompaniment than the PLP feature [13], CRP feature [19], Ellis’s scheme [7], and Serra’s scheme [27]. This phenomenon may also result from the MPLPLC’s ability of representing the singing voice.

³ <http://resources.mpi-inf.mpg.de/MIR/chromatoolbox/>

⁴ <http://labrosa.ee.columbia.edu/projects/coverongs/>

⁵ <http://joanserra.weebly.com/publications.html>

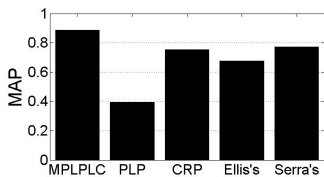


Figure 6. Comparison of robustness against variation in instrumental accompaniments over Collection_2.

3.4 Effect of Cepstrum Mapping and Liftering

To demonstrate the influence of the step LPC cepstrum coefficients mapping and cepstrum liftering on the identification power of the MPLPLC feature, the identification accuracy based on the MPLPLC feature, which is obtained by the MPLPLC model without LPC cepstrum coefficients mapping and cepstrum liftering steps; the MPLPC feature, which is generated by the MPLPLC model without cepstrum liftering step; and the MPLPLC feature, have been compared in terms of TOP-5, MAP, RANK, and MaRR over Collection_1 in Figure 7. It can be seen that both LPC cepstrum coefficients mapping and cepstrum liftering help to enhance the identification power of the MPLPLC feature.

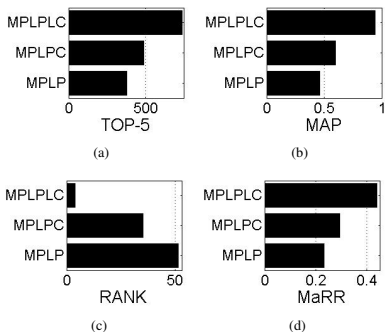


Figure 7. Identification accuracy comparison among MPLP feature, MPLPC feature, and MPLPLC feature, in terms of (a) TOP-5, (b) MAP, (c) RANK, and (d) MaRR over Collection_1.

4. CONCLUSION

We present a new approach, the MPLPLC model, to extract perceptually relevant features from the music signals for pop cover song identification. Here, our main idea is to modify the PLP model, which is a mature technique in the ASR field, by introducing the newest research achievements in psychophysics, such as the time-varying loudness model, auditory filter bank model, and hair cell transduc-

tion model, and by taking the difference between speech and music signals into consideration. Furthermore, LPC cepstrum mapping and cepstrum liftering are combined in the proposed model to boost the resulting feature towards timbre invariance. Experimental results over two music collections show that MPLPLC achieves higher identification accuracy and degree of invariance against instrumental accompaniment than the conventional PLP feature and state-of-the-art music theory based features [7, 19, 27] in the CSI task. This means that the mature techniques in ASR may be modified and used in CSI or other MIR fields.

Despite these achievements, there still exists a lot of room for improvement. Since the MPLPLC feature is based on the modification of PLP, which has been successful in the ASR field, it is good at representing singing voice characteristics. As a result, the MPLPLC-based CSI scheme can identify cover versions with a prominent sing voice very well but not those with only instrumental sounds. To solve this problem, in the near future, we will study the SCI scheme, which is based on the fusion of the MPLPLC feature and the musical facet based features (e.g. PCP-based features), which are good at analyzing harmony-based western music. Furthermore, we plan to look into the application of the MPLPLC feature for other MIR tasks, such as structure analysis, cross-domain music matching, and music segmentation.

5. ACKNOWLEDGEMENT

This work is supported by the National Natural Science Foundation of China (No. 61271349) and the Natural Science Foundation of Shanghai, China (12ZR1415200).

6. REFERENCES

- [1] J. Benesty, M.M. Sondhi, and Y.T. Huang. *Springer Handbook of Speech Processing*. Springer Science & Business Media, 2008.
- [2] S. Bleeck, T. Ives, and R.D. Patterson. Aim-mat: the auditory image model in matlab. *Acta Acustica united with Acustica*, 90(4):781–787, 2004.
- [3] T.M. Chang, E.T. Chen, C.B. Hsieh, and P.C. Chang. Cover song identification with direct chroma feature extraction from aac files. In *2nd Global Conference on Consumer Electronics*, pages 55–56. IEEE, 2013.
- [4] P.J. Clemins and M.T. Johnson. Generalized perceptual linear prediction features for animal vocalization analysis. *The Journal of the Acoustical Society of America*, 120(1):527–534, 2006.
- [5] J.R. Deller, J.G. Proakis, and J.H.L. Hansen. *Discrete-Time Processing of Speech Signals*. IEEE New York, NY, USA:, 2000.
- [6] J.S. Downie. The music information retrieval evaluation exchange (2005-2007): A window into music information retrieval research. *Acoustical Science and Technology*, 29(4):247–255, 2008.

- [7] D.P.W. Ellis and G.E. Poliner. Identifying 'cover songs' with chroma features and dynamic programming beat tracking. In *International Conference on Acoustics, Speech and Signal Processing*, pages IV–1429. IEEE, 2007.
- [8] D.P.W. Ellis and B.M. Thierry. Large-scale cover song recognition using the 2d fourier transform magnitude. In *The 13th International Society for Music Information Retrieval Conference*, pages 241–246, 2012.
- [9] T. Fujishima. Realtime chord recognition of musical sound: A system using common lisp music. In *Proceedings of the International Symposium on Music Information Retrieval*, pages 464–467, 1999.
- [10] B.R. Glasberg and B.C.J. Moore. Derivation of auditory filter shapes from notched-noise data. *Hearing research*, 47(1):103–138, 1990.
- [11] B.R. Glasberg and B.C.J. Moore. A model of loudness applicable to time-varying sounds. *Journal of the Audio Engineering Society*, 50(5):331–342, 2002.
- [12] E. Gómez. *Tonal Description of Music Audio Signals*. PhD thesis, Universitat Pompeu Fabra, 2006.
- [13] H. Hermansky. Perceptual linear predictive (plp) analysis of speech. *Journal of the Acoustical Society of America*, 87(4):1738–1752, 1990.
- [14] B.H. Juang, L. Rabiner, and J.G. Wilpon. On the use of bandpass liftering in speech recognition. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 35(7):947–954, 1987.
- [15] J.C. Junqua, J.P. Haton, and H. Wakita. *Robustness in Automatic Speech Recognition: Fundamentals and Applications*. Kluwer Academic Publishers Boston, USA, 1996.
- [16] J. Makhoul. Spectral linear prediction: Properties and applications. *IEEE Transactions on Acoustics, Speech and Signal Processing*, 23(3):283–296, 1975.
- [17] R. Meddis, M.J. Hewitt, and T.M. Shackleton. Implementation details of a computation model of the inner hair-cell auditory-nerve synapse. *The Journal of the Acoustical Society of America*, 87(4):1813–1816, 1990.
- [18] B.C.J. Moore. Development and current status of the cambridge loudness models. *Trends in hearing*, 18:1–29, 2014.
- [19] M. Müller and S. Ewert. Towards timbre-invariant audio features for harmony-based music. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):649–662, 2010.
- [20] A.V. Oppenheim, R.W. Schaffer, J.R. Buck, and other. *Discrete-Time Signal Processing*, volume 2. Prentice-hall Englewood Cliffs, 1989.
- [21] F. Pachet and J.J. Aucouturier. Improving timbre similarity: How high is the sky. *Journal of Negative Results in Speech and Audio Sciences*, 1(1):1–13, 2004.
- [22] A. Přibilová. Preemphasis influence on harmonic speech model with autoregressive parameterization. *Radioengineering*, 12(3):33–36, 2003.
- [23] J. Salamon. *Melody Extraction from Polyphonic Music Signals*. PhD thesis, Universitat Pompeu Fabra, 2013.
- [24] J. Serrà. *Identification of Versions of the Same Musical Composition by Processing Audio Descriptions*. PhD thesis, Universitat Pompeu Fabra, 2011.
- [25] J. Serrà, E. Gómez, P. Herrera, and X. Serra. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(6):1138–1151, 2008.
- [26] J. Serrà, X. Serra, and R.G. ANDRZEJAK. Cross recurrence quantification for cover song identification. *New Journal of Physics*, 11(9):111–222, 2010.
- [27] J. Serrà, M. Zanin, and R.G. Andrzejak. Cover song retrieval by cross recurrence quantification and unsupervised set detection. *MIREX Extended Abstract*, 2009.
- [28] J.O. Smith. *Mathematics of the Discrete Fourier Transform (DFT): with Music and Audio Applications*. Julius Smith, 2007.
- [29] S.S. Stevens. On the psychophysical law. *Psychological Review*, 64(3):153, 1957.
- [30] H. Terasawa, M. Slaney, and J. Berger. The thirteen colors of timbre. In *Proc. IEEE WASPAA, New Paltz, NY, USA*, pages 323–326, 2005.
- [31] T.C. Walters, D.A. Ross, and R.F. Lyon. The intervalgram: An audio feature for large-scale cover-song recognition. In *From Sounds to Music and Emotions*, pages 197–213. Springer, 2013.
- [32] D.L. Wang and G.J. Brown. *Computational Auditory Scene Analysis: Principles, Algorithms, and Applications*. Wiley-IEEE Press, 2006.
- [33] C. Xiao. Cover song identification using an enhanced chroma over a binary classifier based similarity measurement framework. In *International Conference on Systems and Informatics*, pages 2170–2176. IEEE, 2012.
- [34] J.P. Zbilut, A. Giuliani, and C.L. Webber. Detecting deterministic signals in exceptionally noisy environments using cross-recurrence quantification. *Physics Letters A*, 246(1):122–128, 1998.

RAGA VERIFICATION IN CARNATIC MUSIC USING LONGEST COMMON SEGMENT SET

Shrey Dutta

Dept. of Computer Sci. & Engg.
Indian Institute of Technology
Madras

shrey@cse.iitm.ac.in

Krishnaraj Sekhar PV

Dept. of Computer Sci. & Engg.
Indian Institute of Technology
Madras

pvkrajpv@gmail.com

Hema A. Murthy

Dept. of Computer Sci. & Engg.
Indian Institute of Technology
Madras

hema@cse.iitm.ac.in

ABSTRACT

There are at least 100 *rāgas* that are regularly performed in Carnatic music concerts. The audience determines the identity of *rāgas* within a few seconds of listening to an item. Most of the audience consists of people who are only avid listeners and not performers.

In this paper, an attempt is made to mimic the listener. A *rāga* verification framework is therefore suggested. The *rāga* verification system assumes that a specific *rāga* is claimed based on similarity of movements and motivic patterns. The system then checks whether this claimed *rāga* is correct. For every *rāga*, a set of cohorts are chosen. A *rāga* and its cohorts are represented using pallavi lines of compositions. A novel approach for matching, called Longest Common Segment Set (LCSS), is introduced. The LCSS scores for a *rāga* are then normalized with respect to its cohorts in two different ways. The resulting systems and a baseline system are compared for two partitionings of a dataset. A dataset of 30 *rāgas* from Charsur Foundation¹ is used for analysis. An equal error rate (EER) of 12% is obtained.

1 Introduction

Rāga identification by machine is a difficult task in Carnatic music. This is primarily because a *rāga* is not defined just by the solfege but by *svaras* (ornamented notes) [13]. The melodic histograms obtained for the Carnatic music are more or less continuous owing to the *gamakā*² laden *svaras* of the *rāga* [23]. Although the *svaras* in Carnatic music are not quantifiable, for notational purposes an octave is divided into 12 semitones: S, R1, R2(G1), R3(G2), G3, M1, M2, P, D1, D2(N1), D3(N2) and N3. Each *rāga* is characterised by atleast 5 *svaras*. *Ārohana* and *avarohana* correspond to an ordering of *svaras* in the ascent and de-

¹ <http://www.charsurartsfoundation.org>

² *Gamakā* is a meandering of a *svara* encompassing other permissible frequencies around it.

scend of the *rāga*, respectively. Ragas with linear ordering of *svaras* are referred to as linear ragas such as *Mohonam rāga* (S R2 G3 P D2 S). Similarly, non linear ragas have non linear ordering such as *Ananda Bhairavi rāga* (S G2 R2 G2 M1 P D2 P S). A further complication arises owing to the fact that although the *svaras* in different *rāgas* may be identical, the ordering can be different. Even if the ordering is the same, in one *rāga* the approach to the *svara* can be different, for example, *toḍi* and *dhanyasi*.

There is no parallel in Western classical music to *rāga* verification. The closest that one can associate with, is cover song detection [6, 16, 22], where the objective is to determine the same song rendered by different musicians. Whereas, two different renditions of the same *rāga* may not contain identical renditions of the motifs.

Several attempts have been made to identify *rāgas* [2–4, 7, 8, 12, 14, 26]. Most of these efforts have used small repertoires or have focused on *rāgas* for which ordering is not important. In [26], the audio is transcribed to a sequence of notes and string matching techniques are used to perform *rāga* identification. In [2], pitch-class and pitch-dyads distributions are used for identifying *rāgas*. Bigrams on pitch are obtained using a twelve semitone scale. In [18], the authors assume that an automatic note transcription system for the audio is available. The transcribed notes are then subjected to HMM based *rāga* analysis. In [12, 25], a template based on the *ārohana* and *avarohana* is used to determine the identity of the *rāga*. The frequency of the *svaras* in Carnatic music is seldom fixed. Further, as indicated in [27] and [28], the improvisations in extempore enunciation of *rāgas* can vary across musicians and schools. This behaviour is accounted for in [10, 11, 14] by decreasing the binwidth for computing melodic histograms. In [14], steady note transcription along with n-gram models is used to perform *rāga* identification. In [3] chroma features are used in an HMM framework to perform scale independent *rāga* identification, while in [4] hierarchical random forest classifier is used to match *svara* histograms. The *svaras* are obtained using the Western transcription system. These experiments are performed on 4/8 different *rāgas* of Hindustani music. In [7], an attempt is made to perform *rāga* identification using semi-continuous Gaussian mixtures models. This will work only for linear *rāgas*. Recent research indicates that a *rāga* is characterised best by a time-frequency trajectory rather than a sequence of



	Vocal		Instruments				Total
	Male	Female	Violin	Veena	Saxophone	Flute	
Number of Ragas	25	27	8	3	2	2	30 (distinct)
Number of Artists	53	37	8	3	1	3	105
Number of Recordings	134	97	14	4	2	3	254
Total Duration of Recordings	30 h	22 h	3 h	31 m	10 m	58 m	57 h
Number of Pallavi Lines	655	475	69	20	10	15	1244
Average Duration of Pallavi Lines	11 s	8 s	10 s	6 s	6 s	8 s	8 s (avg.)
Total Duration of Pallavi Lines	2 h	1 h	11 m	2 m	55 s	2 m	3 h

Table 1. Details of the database used. Durations are given in approximate hours (h), minutes (m) or seconds (s).

quantised pitches [5, 8, 9, 19, 20, 24]. In [19, 20], the *sama* of the tala (emphasised by the *bol* of tabla) is used to segment a piece. The repeating pattern in a bandish in Hindustani Khyal music is located using the sama information. In [8, 19], motif identification is performed for Carnatic music. Motifs for a set of five *rāgas* are defined and marked carefully by a musician. Motif identification is performed using hidden Markov models (HMMs) trained for each motif. Similar to [20], motif spotting in an *ālāpana* in Carnatic music is performed in [9]. In [24], a number of different similarity measures for matching melodic motifs of Indian music was attempted. It was shown that the intra pattern melodic motif has higher variation for Carnatic music in comparison with that of Hindustani music. It was also shown that the similarity obtained is very sensitive to the measure used. All these efforts are ultimately aimed at obtaining typical signatures of *rāgas*. It is shown in [9] that there can be many signatures for a given *rāga*. To alleviate this problem in [5], an attempt was made to obtain as many signatures for a *rāga* by comparing lines of compositions. Here again, it was observed that the typical motif detection was very sensitive to the distance measure chosen. Using typical motifs/signatures for *rāga* identification is not scalable, when the number of *rāgas* under consideration increases.

In this paper, this problem is addressed in a different way. The objective is to mimic a listener in a Carnatic music concert. There are at least 100 *rāgas* that are actively performed today. Most listeners identify *rāgas* by referring to the compositions with similar motivic patterns that they might have heard before. In *rāga* verification, a *rāga*'s name (claim) and an audio clip is supplied. The machine has to primarily verify whether the clip belongs to a given *rāga* or not.

This task therefore requires the definition of cohorts for a *rāga*. Cohorts of a given *rāga* are the ragas which have similar movements while at the same time have subtle differences, for example, *darbar* and *nāyaki*. In *darbar* raga, G2 is repeated twice in *avarohana*. The first is more or less flat and short, while the second repetition is inflected. The G2 in *nāyaki* is characterised by a very typical *gamakā*. In order to verify whether a given audio clip belongs to a claimed *rāga*, the similarity is measured with respect to the claimed *rāga* and compared with its cohorts using a novel algorithm called *longest common segment set* (LCSS). LCSS

scores are then normalized using Z and T norms [1, 17].

The rest of the paper is organised as follows. Section 2 describes the dataset used in the study. Section 3 describes the LCSS algorithm and its relevance for *rāga* verification. As the task is *rāga* verification, score normalisation is crucial. Different score normalisation techniques are discussed in Section 4. The experimental results are presented in Section 5 and discussed in Section 6. The main conclusions drawn from the key results in this paper are discussed in Section 7

2 Dataset used

Table 1 gives the details of the dataset used in this work. This dataset is obtained from the Charsur arts foundation³. The dataset consists of 254 vocal and instrument live recordings spread across 30 *rāgas*, including both target ragas and their cohorts. For every new *rāga* that needs to be verified, templates for the *rāga* and its cohorts are required.

2.1 Extraction of pallavi lines

A composition in Carnatic music is composed of three parts, namely, *pallavi*, *anupallavi* and *caranam*. It is believed that the first phrase of the first *pallavi* line of a composition contains the important movements in a *rāga*. A basic sketch is initiated in the *pallavi* line, developed further in the *anupallavi* and *caranam* [21] and therefore contains the gist of the *rāga*. The algorithm described in [21] is used for extracting *pallavi* lines from compositions. Details of the extracted *pallavi* lines are given in Table 1. Experiments are performed on template and test recordings, selected from these *pallavi* lines, as discussed in greater detail in Section 5.

2.2 Selection of cohorts

Wherever possible 4-5 *rāgas* are chosen as cohorts of every *rāga*. The cohorts of every *rāga* were defined by a professional musician. Professionals are very careful about this as they need to ensure that during improvisation, they do not accidentally sketch the cohort. Interestingly, as indicated by the musicians, cohorts need not be symmetric. A *rāga* A can be similar in movement to a *rāga* B , but *rāga* B need not share the same commonality with *rāga* A . The identity of *rāga* B may depend on phrases similar to *rāga* A with some additional movement. For example,

³ <http://www.charsurartsfoundation.org>

to identify the *rāga* Indolam, the phrase G2 M1 D1 N2 S is adequate, while Jayantashree *rāga* requires the phrase G2 M1 D1 N2 S N2 D1 P M1 G2 S.

3 Longest common segment set

In *rāga* verification, matching needs to be performed between two audio clips. The number of similar portions could be more than one and spread across the entire clip. Therefore, there is a need for a matching approach that can find these similar portions without issuing large penalties for gaps in between them. In this section, a novel algorithm called Longest Common Segment Set is described which attempts to do the same.

Let $X = \langle x_1, \dots, x_m; x_i \in \mathbb{R}; i = 1 \dots m \rangle$ be a sequence of m symbols and $Y = \langle y_1, \dots, y_n; y_j \in \mathbb{R}; j = 1 \dots n \rangle$ be a sequence of n symbols where x_i and y_j are the tonic normalized pitch values in cents [9]. The similarity between two pitch values, x_i and y_j , is defined as

$$\text{sim}(x_i, y_j) = \begin{cases} 1 - \frac{|x_i - y_j|^3}{(3s_t)^3} & \text{if } |x_i - y_j| < 3s_t \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where s_t represents a semitone in cents. Due to different styles of various musicians, an exact match between two pitch values contributing to the same *svara* cannot be expected. Hence, in this paper a leeway of 3 semitones is allowed between pitch values. Musically two pitch values, 3 semitones apart, cannot be called similar but this issue is addressed by the cubic nature of the similarity function. The function reaches its half value when the difference in two symbols is approximately half a semitone. Therefore, higher similarity scores are obtained when the corresponding pitch values are at most half a semitone apart.

A common subsequence Z_{XY} in sequences X and Y is defined as

$$Z_{XY} = \begin{cases} \langle (x_{i_1}, y_{j_1}), \dots, (x_{i_p}, y_{j_p}) \rangle \\ 1 \leq i_1 < \dots < i_p \leq m \\ 1 \leq j_1 < \dots < j_p \leq n \\ \text{sim}(x_{i_k}, y_{j_k}) \geq \tau_{\text{sim}} \end{cases} \quad (2)$$

where τ_{sim} is a threshold which decides the membership of the symbol pair (x_{i_k}, y_{j_k}) in a subsequence Z_{XY} . The value of τ_{sim} is decided empirically based on the domain of the problem as discussed in Section 5. An example common subsequence is shown with red color in Figure 1.

3.1 Common segments

Continuous symbol pairs in a common subsequence are referred to as a segment. Two different types of segments are defined, namely hard and soft segments.

Hard segment is a group of common subsequence symbols such that there are no gaps in between as shown in green color in Figure 1. Then a hard segment, starting with

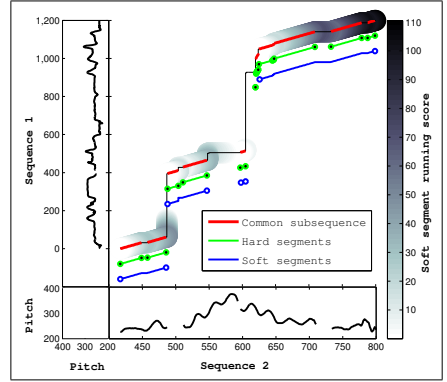


Figure 1. An example of a common segment set between two sequences representing the real data

a symbol pair (x_i, y_j) , must be of the form

$$H_{X_i Y_j}^l = \begin{cases} \langle (x_i, y_j), (x_{i+1}, y_{j+1}), \dots, (x_{i+l}, y_{j+l}) \rangle \\ 1 \leq i < i+1 < \dots < i+l \leq m \\ 1 \leq j < j+1 < \dots < j+l \leq n \end{cases} \quad (3)$$

where $l+1$ represents the length of the hard segment. The score of the k^{th} hard segment $H_{X_{i_k} Y_{j_k}}^l$ is defined as

$$\text{hc}(H_{X_{i_k} Y_{j_k}}^l) = \sum_{d=0}^l \text{sim}(x_{i_k+d}, y_{j_k+d}) \quad (4)$$

Soft segment is a group of common subsequence symbols where gaps are permitted with a penalty. Therefore, a soft segment consists of one or more hard segments (shown with blue color in Figure 1). The gaps between the hard segments decides the penalty assigned. Thus, the score of the k^{th} soft segment $S_{X_{i_k} Y_{j_k}}$, consisting of r hard segments, is defined as

$$\text{sc}(S_{X_{i_k} Y_{j_k}}) = \sum_{s=1}^r \text{hc}(H_{X_{i_k} Y_{j_k}}^l) - \gamma\rho \quad (5)$$

where γ is the total number of gaps between r hard segments and ρ is the penalty for each gap. The number of hard segments to be included in a soft segment is decided by the running score of the soft segment. The running score of the soft segment increases during the hard segment and decreases during the gap due to penalties as shown in gray-scale in Figure 1. During a gap, if the running score decreases below a threshold τ_{rc} (or becomes almost white in Figure 1) then that gap is ignored and all the hard segments, encountered before it, are included into a soft segment.

3.2 Common segment set

All segments together correspond to a segment set. The score of a segment set (ss) is defined as

$$score(ss_{XY}) = \frac{\sum_{k=1}^p c(Z_{X_i, Y_{j_k}})^2}{\min(m, n)^2} \quad (6)$$

where p is the number of segments, c refers to the score computed in either (4) or (5) and Z refers to a segment (hard or soft). This equation gives preference to longer segments. For example, in case 1, there are 10 segments each of length 2 and in case 2, there are 4 segments each of length 5. In both the cases the total length of the segments is 20 but in (6), case 1 is scored as 0.1 and case 2 is scored as 0.25 when the denominator is taken to be 20^2 . Longer matched segments could be considered as a phrase or an essential part of it. Whereas, shorter matched segments could generally mean noise. Therefore, there is a heavier penalty for shorter segments.

3.3 Longest common segment set

Longest common segment set (lcss) is a segment set with maximum score value as defined in (7).

$$lcss_{XY} = \underset{ss_{XY}}{argmax} (score(ss_{XY})) \quad (7)$$

Therefore, lcss can be obtained by maximizing score in (6) using dynamic programming.

3.4 Dynamic Programming algorithm to find longest common segment set

The algorithm for finding the optimum soft segment set is given in Algorithm 1. Optimum hard segment sets are found similarly. In the algorithm, tables c and s are used for storing the running score and the score of the common segment sets, respectively. Table a is used for storing the partial scores from s . Table d is maintained for backtracking the path of the LCSS. The arrows represent the subpath to take while backtracking (up, left or cross). Input sequences to function LCSS are appended with symbols ϕ_x and ϕ_y such that their similarity with any symbol is 0. This is mainly required to compute the last row and column of score table. On similarity, line 8 updates the running score with a value based on the similarity, whereas line 9 updates the score using the previous diagonal entry. When symbols are dissimilar a gap is found. Lines 12 and 19 are used to penalize the running score. If it is an end of the segment then line 14 and 21 updates score as per (6). Line 26 updates table a with the score value of the current segment set when the beginning of a new segment is encountered. When a gap is encountered line 28 updates it to -1 . To find the longest common segment set, backtracking is performed to obtain the path in table d that has the maximum score as given by table s . The boundaries of soft segments can be found using the cost values while tracing the path.

4 Raga Verification

Let $T_{raga} = \{t_1, t_2, \dots, t_{N_{raga}}\}$ represent a set of template recordings, where 'raga' refers to the name of the

Algorithm 1 Algorithm for Soft-Longest Common Segment Set

Data:

c - table of size $(m+2) \times (n+2)$ for storing running score
 s - table of size $(m+2) \times (n+2)$ for storing score
 d - table of size $(m+2) \times (n+2)$ for path tracking
 a - table of size $(m+2) \times (n+2)$ for storing partial scores.

```

1: function LCSS ( $\langle x_1, \dots, x_m, \phi_x \rangle, \langle y_1, \dots, y_n, \phi_y \rangle$ )
2:   Initialize 1st row and column of  $c$ ,  $s$ ,  $d$  and  $a$  to 0
3:    $p \leftarrow \min(m, n)$ 
4:   for  $i \leftarrow 1$  to  $m+1$  do
5:     for  $j \leftarrow 1$  to  $n+1$  do
6:       if  $\text{sim}(x_i, y_j) > \tau_{sim}$  then
7:          $d_{i,j} \leftarrow \swarrow$ 
8:          $c_{i,j} \leftarrow c_{i-1, j-1} + \left( \frac{\text{sim}(x_i, y_j) - \tau_{sim}}{1 - \tau_{sim}} \right)$ 
9:          $s_{i,j} \leftarrow s_{i-1, j-1}$ 
10:      else if  $c_{i-1, j} < c_{i, j-1}$  then
11:         $d_{i,j} \leftarrow \uparrow$ 
12:         $c_{i,j} \leftarrow \max(c_{i-1, j} - \rho, 0)$ 
13:      if  $d_{i-1, j} = \swarrow$  then
14:         $s_{i,j} \leftarrow \frac{a_{i-1, j} * p^2 + c_{i-1, j}^2}{p^2}$ 
15:      else
16:         $s_{i,j} \leftarrow s_{i-1, j}$ 
17:      else
18:         $d_{i,j} \leftarrow \leftarrow$ 
19:         $c_{i,j} \leftarrow \max(c_{i, j-1} - \rho, 0)$ 
20:      if  $d_{i, j-1} = \swarrow$  then
21:         $s_{i,j} \leftarrow \frac{a_{i, j-1} * p^2 + c_{i, j-1}^2}{p^2}$ 
22:      else
23:         $s_{i,j} \leftarrow s_{i, j-1}$ 
24:       $q \leftarrow \max(a_{i-1, j-1}, a_{i-1, j}, a_{i, j-1})$ 
25:      if  $q = -1$  and  $d_{i,j} = \swarrow$  then
26:         $a_{i,j} \leftarrow s_{i-1, j-1}$ 
27:      else if  $c_{i,j} < \tau_{rc}$  then
28:         $a_{i,j} \leftarrow -1$ 
29:      else
30:         $a_{i,j} \leftarrow q$ 

```

raga and N_{raga} is the total number of templates for that raga. During testing, an input test recording, X , with a claim is tested against all the template recordings of the claimed raga. The final score is computed as given in (8).

$$score(X, \text{claim}) = \max_{Y \in T_{claim}} (score(lcss_{XY})) \quad (8)$$

The final decision, of accepting or rejecting the claim, directly based on this score could be erroneous. Score normalisation with cohorts is essential to make a decision, especially when the difference between two ragas is subtle.

4.1 Score Normalization

LCSS scores corresponding to correct and incorrect claims are referred as true and imposter scores, respectively. If the imposter is a cohort raga, then the imposter score is also referred as cohort score. Various score normalization techniques are discussed in the literature for speech recog-

nition, speaker/language verification and spoken term detection [1, 17].

Zero normalization (Z -norm) uses the mean and variance estimate of cohort scores for scaling. The advantage of Z -norm is that the normalization parameters can be estimated off-line. Template recordings of a *rāga* are tested against template recordings of its cohorts and the resulting scores are used to estimate a *rāga* specific mean and variance for the imposter distribution. The normalized scores using Z -norm can be calculated as

$$\text{score}_{\text{norm}}(X, \text{claim}) = \frac{\text{score}(X, \text{claim}) - \mu_i^{\text{claim}}}{\sigma_i^{\text{claim}}} \quad (9)$$

where μ_i^{claim} and σ_i^{claim} are the estimated imposter parameters for the claimed *rāga*.

Test normalization (T -norm) is also based on a mean and variance estimation of cohort scores for scaling. The normalization parameters in T -norm are estimated online as compared to their offline estimation in Z -norm. During testing, a test recording is tested against template recordings of cohort *rāgas* and the resulting scores are used to estimate mean and variance parameters. These parameters are then used to perform the normalization given by (9).

The test recordings of a *rāga* may be scored differently against templates corresponding to the same *rāga* or imposter *rāga*. This can cause overlap between the true and imposter score distributions. T -norm attempts to reduce this overlap. The templates that are stored and the audio clip that is used during test can be from different environments.

5 Performance evaluation

In this section, we describe the results of *rāga* verification using LCSS algorithm in comparison with Rough Longest Common Subsequence (RLCS) algorithm [15] and Dynamic Time Warping (DTW) algorithm using different normalizations.

5.1 Experimental configuration

Only 17 *rāgas* out of 30 were used for *rāga* verification as only for 17 *rāgas* sufficient number of relevant cohorts could be obtained from the 30 *rāgas*. This is due to non-symmetric nature of the cohorts as discussed in Section 2. For *rāga* verification, 40% of the pallavi lines are used as templates and remaining 60% are used for testing. This partitioning of dataset is done into two ways, referred as D1 and D2. In D1, the variations of a pallavi line might fall into both templates and test though it is not necessary. Variations of a pallavi line are different from the pallavi line due to improvisations. In D2, these variations can either belong to template or they all belong to test but strictly not present in both. The values of thresholds τ_{sim} and τ_{rc} are empirically chosen as 0.45 and 0.5, respectively. Penalty, ρ , issued for gaps in segments is empirically chosen as 0.5.

5.2 Results

Table 2 and Figure 2 show the comparison of LCSS with DTW and RLCS using different normalizations. Equal Er-

Algorithm	Dataset	No Norm	Z -norm	T -Norm
DTW	D1	27.78	29.88	17.45
	D2	40.81	40.03	35.96
RLCS	D1	24.43	27.22	14.87
	D2	41.72	42.58	41.20
LCSS (hard)	D1	29.00	31.75	15.65
	D2	40.28	40.99	34.11
LCSS (soft)	D1	21.89	24.11	12.01
	D2	37.24	38.96	34.57

Table 2. EER(%) for different algorithms using different normalizations on different datasets.

ror rate (EER) refers to a point where false alarm rate and miss rate is equal. For T -norm, the best 20 cohort scores were used for normalization. LCSS (soft) with T -norm performs best for D1 around the EER point, and for high miss rates and low false alarms, whereas it performs poorer than LCSS (hard) for low miss rates and high false alarms. This behavior appears to be reversed for D2. The magnitude around EER is much greater for D2. This is because, none of the variations of the pallavi lines in test are present in the templates. It is also shown that RLCS performs poorer than any other algorithms for D2. The curves also show no improvements for Z -norm compared to baseline with no normalization. This can happen due to the way normalization parameters are estimated for Z -norm. For example, some of the templates, which may not be similar to the test, can be similar to some of the cohorts' templates, resulting in higher mean. This would not have happened in T -norm where the test itself is tested against the cohorts' templates.

6 Discussion

In this section, we discuss how LCSS (hard) and LCSS (soft) can be combined to achieve better performance. We also verify that T -norm reduces the overlap between true and imposter scores.

6.1 Combining hard-LCSS and soft-LCSS

Instead of selecting a threshold, we will assume that a true claim is correctly verified when its score is greater than all the cohort scores. Similarly, a false claim is correctly verified when its score is lesser than atleast one of the cohort scores. Table 3 shows the number of claims correctly verified only by hard-LCSS, only by soft-LCSS, by both and by neither of them. It is clear that there is an overlap between the correctly verified claims of hard-LCSS and soft-LCSS. Nonetheless, the number of claims distinctly verified by both is also significant. Therefore, the combination of these two algorithms could result in a better performance.

6.2 Reduction of overlap in score distribution by T -norm

Figure 3 shows the effect of T -norm on the distribution of hard-LCSS scores. It is clearly seen that the overlap, between the true and imposter score distributions, is reduced

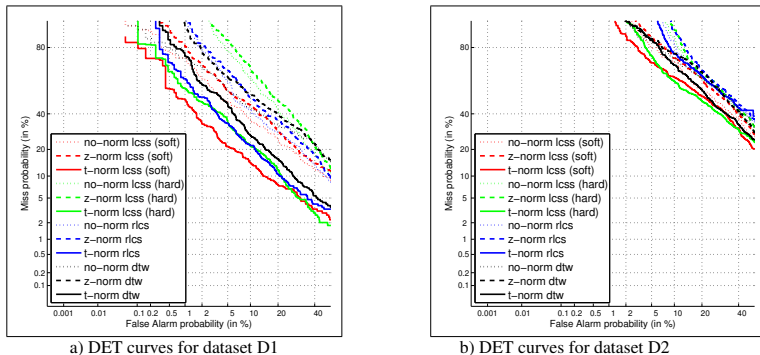


Figure 2. DET curves comparing LCSS algorithm with different algorithms using different score normalizations

Dataset	Claim-type	Hard-only	Soft-only	Both	Neither
D1	True	23	55	289	77
	False	46	78	1745	54
D2	True	47	23	155	220
	False	99	75	1585	168

Table 3. Number of claims correctly verified by hard-LCSS only, by soft-LCSS only, by both and by neither of them for D1 and D2 using T -norm

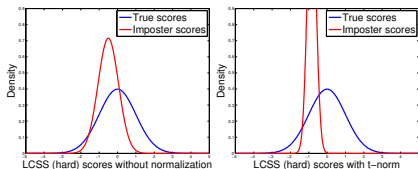


Figure 3. Showing the effect of T -norm on the score distribution

significantly. For visualization purposes, the true score distributions are scaled to zero mean and unit variance and corresponding imposter score distributions are scaled appropriately.

6.3 Scalability of $rāga$ verification

The verification of a $rāga$ depends on the number of its cohort $rāgas$ which are usually 4 or 5. Since it does not depend on all the $rāgas$ in the dataset, as in $rāga$ identification, any number of $rāgas$ can be added to the dataset.

7 Conclusion and future work

In this paper, we have proposed a different approach to $rāga$ analysis in Carnatic music. Instead of $rāga$ identi-

fication, $rāga$ verification is performed. A set of cohorts for every $rāga$ is defined. The identity of an audio clip is presented with a claim. The claimed $rāga$ is verified by comparing with the templates of the claimed $rāga$ and its cohorts by using a novel approach. A set of 17 $rāgas$ and its cohorts constituting 30 $rāgas$ is tested using appropriate score normalization techniques. An equal error rate of about 12% is achieved. This approach is scalable to any number of $rāgas$ as the given $rāga$ and its cohorts need to be added to the system.

8 Acknowledgments

This research was partly funded by the European Research Council under the European Unions Seventh Framework Program, as part of the CompMusic project (ERC grant agreement 267583). We are grateful to Padmasundari for selecting the cohorts.

9 References

- [1] R. Auckenthaler, M Carey, and H Lloyd-Thomas. Score normalisation for text-independent speaker verification systems. *Digital Signal Processing*, 10:42–54, 2000.
- [2] P Chordia and A Rae. Raag recognition using pitch-class and pitch-class dyad distributions. In *Proceedings of International Society for Music Information Retrieval Conference*, pages 431–436, 2007.
- [3] Pranay Dighe, Parul Agarwal, Harish Karnick, Sid-dhartha Thota, and Bhiksha Raj. Scale independent raga identification using chromagram patterns and swara based features. In *Proceedings of IEEE International Conference on Multimedia and Expo Workshops*, pages 1–4, 2013.
- [4] Pranay Dighe, Harish Karnick, and Bhiksha Raj. Swara histogram based structural analysis and identification of indian classical ragas. In *Proceedings of 14th International Society for Music Information Retrieval Conference*, pages 35–40, 2013.

- [5] Shrey Dutta and Hema A. Murthy. Discovering typical motifs of a raga from one-liners of songs in carnatic music. In *Proceedings of 15th International Society for Music Information Retrieval Conference*, pages 397–402, 2014.
- [6] D.P.W. Ellis and G.E. Poliner. Identifying ‘cover songs’ with chroma features and dynamic programming beat tracking. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing*, volume 4, pages IV–1429–IV–1432, 2007.
- [7] S Arthi H G Ranjani and T V Sreenivas. Shadja, swara identification and raga verification in alapana using stochastic models. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 29–32, 2011.
- [8] Vignesh Ishwar, Ashwin Bellur, and Hema A Murthy. Motivic analysis and its relevance to raga identification in carnatic music. In *2nd CompMusic Workshop*, 2012.
- [9] Vignesh Ishwar, Shrey Dutta, Ashwin Bellur, and Hema A. Murthy. Motif spotting in an alapana in carnatic music. In *Proceedings of 14th International Society for Music Information Retrieval Conference*, pages 499–504, 2013.
- [10] Gopala Krishna Koduri, Sankalp Gulati, and Preeti Rao. A survey of raaga recognition techniques and improvements to the state-of-the-art. *Sound and Music Computing*, 2011.
- [11] Gopala Krishna Koduri, Sankalp Gulati, Preeti Rao, and Xavier Serra. Raga recognition based on pitch distribution methods. *Journal of New Music Research*, 41(4):337–350, 2012.
- [12] A.S. Krishna, P.V. Rajkumar, K.P. Saishankar, and M. John. Identification of carnatic raagas using hidden markov models. In *IEEE 9th International Symposium on Applied Machine Intelligence and Informatics*, pages 107–110, 2011.
- [13] T M Krishna and Vignesh Ishwar. Carnatic music : Svara, gamaka, motif and raga identity. In *2nd Comp-Music Workshop*, 2012.
- [14] V. Kumar, H. Pandya, and C.V. Jawahar. Identifying ragas in indian music. In *Proceedings of 22nd International Conference on Pattern Recognition*, pages 767–772, 2014.
- [15] Hwei-Jen Lin, Hung-Hsuan Wu, and Chun-Wei Wang. Music matching based on rough longest common subsequence. *Journal of Information Science and Engineering*, pages 27, 95–110., 2011.
- [16] Meinard Müller, Frank Kurth, and Michael Clausen. Audio matching via chroma-based statistical features. In *Proceedings of International Conference on Music Information Retrieval*, pages 288–295, 2005.
- [17] Jiri Navratil and David Klusacek. On linear dets. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 229–232, 2007.
- [18] Gurav Pandey, Chaitanya Mishra, and Paul Ipe. Tansen : A system for automatic raga identification. In *Proceedings of 1st Indian International Conference on Artificial Intelligence*, pages 1350–1363, 2003.
- [19] P. Rao, J. Ch. Ross, K. K. Ganguli, V. Pandit, V. Ishwar, A. Bellur, , and H. A. Murthy. Melodic motivic analysis of indian music. *Journal of New Music Research*, 43(1):115–131, 2014.
- [20] Joe Cheri Ross, Vinutha T. P., and Preeti Rao. Detecting melodic motifs from audio for hindustani classical music. In *Proceedings of 13th International Society for Music Information Retrieval Conference*, pages 193–198, 2012.
- [21] Sridharan Sankaran, Krishnaraj P V, and Hema A Murthy. Automatic segmentation of composition in carnatic music using time-frequency cfcc templates. In *Proceedings of 11th International Symposium on Computer Music Multidisciplinary Research*, 2015.
- [22] J. Serra, E. Gomez, P. Herrera, and X. Serra. Chroma binary similarity and local alignment applied to cover song identification. *IEEE Transactions on Audio, Speech, and Language Processing*, 16(6):1138–1151, 2008.
- [23] Joan Serrà, Gopala K. Koduri, Marius Miron, and Xavier Serra. Assessing the tuning of sung indian classical music. In *Proceedings of 12th International Society for Music Information Retrieval Conference*, pages 157–162, 2011.
- [24] Sankalp Gulati Joan Serra and Xavier Serra. An evaluation of methodologies for melodic similarity in audio recordings of indian art music. In *Proceedings of 40th IEEE International Conference on Acoustics, Speech and Signal Processing*, 2015.
- [25] Surendra Shetty. Raga mining of indian music by extracting arohana-avarohana pattern. *International Journal of Recent trends in Engineering*, 1(1), 2009.
- [26] Rajeswari Sridhar and Tv Geetha. Raga identification of carnatic music for music information retrieval. *International Journal of Recent trends in Engineering*, 1(1):1–4, 2009.
- [27] M Subramanian. Carnatic ragam thodi pitch analysis of notes and gamakams. *Journal of the Sangeet Natak Akademi*, XLI(1):3–28, 2007.
- [28] D Swathi. Analysis of carnatic music: A signal processing perspective. MS Thesis, IIT Madras, India, 2009.

INSTRUMENT IDENTIFICATION IN OPTICAL MUSIC RECOGNITION

Yucong Jiang

School of Informatics and Computing
Indiana University, Bloomington
yujiang@indiana.edu

Christopher Raphael

School of Informatics and Computing
Indiana University, Bloomington
craphael@indiana.edu

ABSTRACT

We present a method for recognizing and interpreting the text labels for the instruments in an orchestra score, thereby associating staves with instruments. This task is one of many necessary in optical music recognition. Our approach treats the score system as the basic unit of processing. A graph structure describes the possible orderings of instruments in the system. Each instrument may apply to several staves, may be represented with several possible text strings, and may appear at several possible positions relative to the staves. We find the optimal labeling of staves using a globally optimal dynamic programming approach that embeds simple template-based optical character recognition within the overall recognition scheme. When given an entire score, we simultaneously optimize on the text labeling for each system, as well as the character template models, thus adapting to the font at hand. Our implementation alternately optimizes over the text label identification and re-estimates the character templates. Experiments are presented on 10 different scores showing a significant improvement due to adaptation.

1. INTRODUCTION

In some scores, particularly those for small ensemble, instruments appear in the same position in all systems making it easy to associate instruments with staves. This scheme would be typical for a string quartet or sonata for solo instrument and piano. Occasionally large-ensemble scores follow this convention as well, though it requires considerably more space as empty staves must be written out every time any instrument is not used in a particular system, thus creating longer scores and lowering the density of information. For these reasons many publishers avoid this layout style, instead notating only the instruments that play in a particular system. In this case text labels, usually appearing in the left margin of the system, identify the instrument(s) associated with the individual staves, as in Figure 1. These are the scores we treat here, while our goal is the labeling of each staff with its associated instrument. Such labeling is necessary for nearly any aspect of optical music

recognition (OMR) using “instrument-labeled scores”, as it allows one to link systems together in a meaningful way.

The first steps of our OMR system [9] are to identify the staves in a page, and then to group these into systems. While these tasks present challenges due to the wide variation in printed scores, they are among the easier OMR tasks, and are handled reasonably well by our system. The resulting score systems, including the precise locations of all the staves they contain, constitute the input to our staff labeling process.

In spite of the mature nature of optical character recognition (OCR), our staff labeling problem is highly challenging when viewed purely in these terms. The text strings we seek to recognize are usually a single word or abbreviation, thus providing only a small portion of data for each recognition problem. Furthermore, even though the vocabulary is constrained to the instrument names used in the score, OCR will encounter difficulties distinguishing similar strings, such as “Violin I” and “Violin II,” or “Vln.,” “Vla.,” and “Vcl.,”. Finally, there often is other irrelevant text in the area of the names we seek to recognize, further hindering the recognition. But even if OCR were enough to recognize the instrument names, our goal includes more than this. As it is common for text strings to apply to multiple staves in a score, we need the name-to-staff mapping as well.

Thus, unlike the bottom-up approach used in [12], our top-down approach uses a graphical model that generates all legitimate possible partitions of the system into staves and all legitimate possible labelings of these partitions with instruments. The graphical representation enables a dynamic programming approach that embeds rather generic OCR into the “innermost loop” for the recognition of individual text labels. The model may include strong assumptions about the possible orderings of instruments, with the most obvious choice being that the instruments appear in the order initially given on the first score page, (though any subset of instruments can be omitted).

Perhaps the biggest challenge of our task is the font variation between scores. Even controlling for the height of the staff, one still sees considerable variation in the size and shape of the characters between fonts. It might be possible to develop an *omnifont* approach [2,3], meaning a text model that is trained from a variety of fonts, and thus capable of recognizing this same variety. However, as character models are required to accept a wider range of presentations for each given letter, they become less capable of dis-



© Yucong Jiang, Christopher Raphael.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Yucong Jiang, Christopher Raphael. “Instrument Identification in Optical Music Recognition”, 16th International Society for Music Information Retrieval Conference, 2015.



Figure 1. One page with an 8 staves system and an 11 staves system.

tinguishing between different letters. For this reason omnifont models generally perform worse than models tuned for the specific task at hand.

Borrowing a well-known idea from pattern recognition, [11, 13], we address this challenge by *simultaneously* recognizing the instrument labels on the entire collection of systems in a score *and* learning the font model for the document at hand. Our algorithm iteratively recognizes the systems, then retrains the font models using the optimal labeling and text alignments produced in the recognition phase. One might expect that this approach is simply too greedy to succeed, failing to explore the high-dimensional world of possible character models and system interpretations, while almost guaranteed to get stuck in a mediocre local optimum. However, we present experiments that show a large *monotonic* improvement in recognition accuracy as we iterate this process, culminating with excellent recognition results. The approach is feasible due to the highly restrictive assumption made by our graphical model, thus constraining the admissible interpretations to a tiny fraction of those arising without this restriction. Our experiments demonstrate that this graph model is the difference between basically successful and unsuccessful results.

2. LABELING STAVES WITH INSTRUMENTS

The usual notational convention for large-ensemble scores lists all instruments on the first page of a piece or movement, whether or not the instruments play in this page [10].

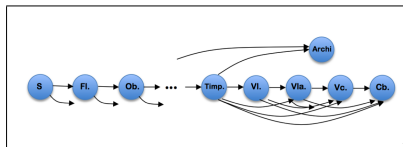


Figure 2. A directed graph representing the possible orderings for the instruments.

In subsequent pages, instruments, perhaps in abbreviated form, are written in the left score margin. Usually the instrument labels are displayed immediately to the left of the associated staff line, though the labels sometimes describe *collections* of staves, such as “Strings”, “Horns”, etc. In such a case the text label usually appears centered with respect to the group of staves, often emphasized by a bracketing of the associated staves in the left margin of the score. Figure 1 shows a typical example.

2.1 THE MODEL

Our staff labeling procedure requires input from the user explaining the labeling scheme(s) used in the score at hand. Typically, the instruments appearing in a system are a subsequence of the order given on the first page of the score. However, variations are possible such as substituting a collective name for the individual labels, e.g. using the single text label “Strings” instead of the individual labels “Violin 1”, “Violin 2”, “Viola”, “Violoncello” and “Bass” [10]. We assume that the possible labelings can be described by a directed graph, G , as in Figure 2, where the possible paths through the graph give the legitimate label sequences. We assume the graph is supplied by the user either implicitly or explicitly. Each vertex $g \in G$ is associated with an instrument, $I(g)$, so recovering the correct path will give the sequence of instruments employed in the system.

As mentioned above there may be several staves associated with a particular instrument or group of instruments, though we do not require such a convention to be followed consistently. We rely on the user to list, for each instrument, the possible labeling variations encountered in the score. We describe this information as a collection of *patterns* for each graph node, $P(g) = \{p_1, \dots, p_c\}$, where we suppress the dependence of the list length, c , on g in our notation. Each pattern, p has three attributes: $p = (p^k, p^l, p^a)$ giving the number of staves used for the instrument(s), p^k , the location of the text label with respect to the group of staves, p^l , and the specific character sequence used for the text label, p^a . For instance $p^l = 0$ would mean that the label appears in the middle of the group of staves (next to the middle staff if p^k is odd and between the middle two staves if p^k is even). When $p^l \neq 0$, p^l gives the integral number of inter-staff half spaces above or below the middle location where the text will be found. In Figure 1 $p^l = 0$ for all instruments.

Every pair of adjacent vertices, g', g are connected by

$|P(g)|$ directed arcs labeled with the various patterns, $P(g)$, though these are not explicitly drawn in Figure 2. Thus there may be several arcs that connect g' to g , each accounting for a possible number of staves for the instrument, $I(g)$, location of the text label, and the actual text itself. There is a one-to-one correspondence between the allowable labelings of each system and the legal paths through the graph. That is, if s, g_1, \dots, g_M is a path beginning from the start vertex, s , with arc labels p_1, \dots, p_M that correctly accounts for the number of staves in the system, N ,

$$\sum_{m=1}^M p_m^k = N \quad (1)$$

then the labeling associates the first p_1^k staves with instrument $I(g_1)$, the next p_2^k staves with instrument $I(g_2)$ and so on. The path may terminate anywhere in the graph other than at the start vertex, s , as long as Eqn. 1 is satisfied.

2.2 RECOGNIZING THE TEXT LABELING

We score every legal path through G as a sum of arc scores and compute the best scoring path through dynamic programming (DP). For this purpose we define $E(n, g)$, for $n = 0, \dots, N$, $g \in G$, as the best scoring interpretation of the first n staves ending in state g . We compute E by initializing $E(0, s) = 0$, then visiting the staves in order: $n = 1, \dots, N$ computing for each $g \in G$,

$$E(n, g) = \max_{g' \xrightarrow{p^k} g} E(n - p^k, g') + L(n, p) \quad (2)$$

where the maximum is over all legal arcs going from g' to g with $p^k \leq n$. In Eqn. 2 $L(n, p)$ is the arc score measuring the plausibility that the text label p^α positioned at relative position p^l is used for staves $n - p^k + 1, \dots, n$. While we present L in more detail in Section 2.3, for now it suffices to say that L measures the quality of the best match of the text, p^α , to the score image data in the area determined by n, p^k, p^l . It is worth noting that $L = 0$ is a neutral result, meaning that the optimal placement of the letters of p^α explains the data as well as a background model. In contrast, positive (negative) scores of L indicate evidence for (against) the labeling implied by the transition of $g' \xrightarrow{p^k} g$. Thus our algorithm has no inherent bias for assigning more or less text labels in the optimal interpretation.

Having computed $E(n, g)$ for $n = 1, \dots, N$ and $g \in G$, the score of the optimal path is given by $\max_{g \in G} E(N, g)$, while it is a simple matter to recover the optimal sequence of vertices and transitions that produce the optimal score. We denote these by g_1^*, \dots, g_M^* and p_1^*, \dots, p_M^* .

2.3 CHARACTER RECOGNITION

Our approach to character recognition is standard template-based [7, 8], and will only be discussed briefly and informally here. $L(n, p)$ evaluates the quality of the hypothesis n, p . The information contained in n and p collectively describes a reasonably precise vertical location in the image. The task in computing L is to search the area around this

subject for the optimal locations of the characters of p^α , subject to reasonable constraints regarding their spacing.

Suppose the characters of p^α : c_1, \dots, c_L have rectangular templates m_1, \dots, m_L which are hypothesized to be placed at image locations $(x_1, y_1), \dots, (x_L, y_L)$. Each template is a matrix of values $m_i(i, j) \in \mathcal{M}$ where $\mathcal{M} = \{b, w, t, n\}$ indexes *black*, *white*, *transitional*, and *null* grey level probability models denoted by P_b, P_w, P_t, P_n . P_b mostly “expects” to see low grey levels, P_w “expects” to see high grey levels, P_t is a mixture of these two models, and P_n is a *null* or *background* model taken as the normalized grey level histogram of the entire image. $L_{x,y}(n, p)$ is then defined to be the normalized data log likelihood given by

$$L_{x,y}(n, p) = \sum_{i=1}^L \sum_{i,j} \log \frac{P_{m_i(i,j)}(J(x_i + i, y_l + j))}{P_n(J(x_i + i, y_l + j))} \quad (3)$$

where (x, y) denotes the entire collection of template locations, $J(x, y)$ is the image grey level at pixel (x, y) and the inner sum uses the range for i, j appropriate for the l th rectangular character template.

In computing Eqn. 3 we consider a variety of possible vertical positions for the text baseline, and all reasonable positions for the characters along that baseline so that

$$L(n, p) = \max_{x,y} L_{x,y}(n, p).$$

Thus the computation consists of a loop over baseline positions with each iteration accomplished by a DP computation that optimally locates the character templates.

2.4 SIMULTANEOUS OPTIMIZATION

Section 2.2 gives our procedure for finding the optimal text labeling for the staves of a system. Computing this labeling requires at least reasonable character templates, though it would be preferable to have templates that represent the font at hand. Unfortunately, fonts differ greatly from one music document to another, both in size and shape, so we have no way of knowing *a priori* the font used for instrument names in any given score. Our approach here is to *simultaneously* estimate both the optimal text labeling and the optimal character templates, thus *adapting* to the font at hand while we recognize. While simultaneous estimation of both interpretation and model parameters is infeasible for many recognition problems, we rely here on the strong graph-based assumptions we have made on the family of possible labelings. In essence, our assumptions about instrument order are powerful enough to get reasonable estimates of the instrument labels even with poorly specified character templates. Thus we can use this labeling, and the precise character template positions that come with it, to re-estimate our character templates. Our overall approach then becomes an iteration between the (re)estimation of instrument labels and the (re)estimation of character templates, similar with [5, 6]. In practice this approach converges after only a few iterations, and usually does so with significantly better recognition accuracy than with the original character templates, as discussed in Section 3.

More precisely, we let z denote a possible text labeling for the entire collection of system staves. Thus z includes a path of vertices and arcs through G for each system in the score, as discussed in Section 2.2, as well as the character template positions that result from the $L(n, p)$ computations of Section 2.3. Let θ denote the complete collection of character templates employed in Section 2.3, for all letters and punctuation used in the text labels. Finally we let $\bar{E}(z, \theta)$ be the summed data log likelihood score of Eqn. 3 produced by evaluating the complete set of recognized characters in z at their precise positions using the character templates of θ .

Starting from our initial character templates θ_0 , the basic two-stage iteration of our algorithm is then

$$z_{l+1} = \arg \max_z E(z, \theta_l) \quad (4)$$

$$\theta_{l+1} = \arg \max_{\theta} E(z_{l+1}, \theta) \quad (5)$$

for $l = 0, 1, \dots$. The update for z in Eqn. 4 is simply the dynamic programming procedure from Section 2.2 applied to each system in the score, which is guaranteed to produce a global optimum for z . The θ update is accomplished by maximum likelihood estimation, as follows. Suppose our alphabet of characters and punctuation is c^1, \dots, c^Q . Also suppose that c^q appears at locations $(x_1^q, y_1^q), \dots, (x_R^q, y_R^q)$ as defined through the text labeling and implicit template alignment of Section 2.2. We then let

$$c^q(i, j) = \arg \max_{\mu \in \mathcal{M}} \sum_{r=1}^R \log \frac{P_n(J(x_r^q + i, y_r^q + j))}{P_n(J(x_r^q + i, y_r^q + j))}$$

for $q = 1, \dots, Q$, which is, by definition, Eqn. 5.

The proposed algorithm is simply coordinate-wise optimization over z and θ , which guarantees that the sequence $E(z_1, \theta_1), E(z_2, \theta_2) \dots$ is non-decreasing. Furthermore, the sequence is guaranteed to converge due to our finite (but large) domain. In practice, this happens in only a few iterations.

It is worth noting that our strategy is different from the usual EM scheme [4] for performing maximum likelihood estimation of model parameters. To implement EM we would need a probabilistic model for the graph transitions, which would be easy to supply, but absent from our current formulation. However, EM attempts to increase the *marginal* data likelihood rather than the likelihood of the optimal path. Thus, if we were to replace our parameter estimation step by an iteration of EM, we still may end up decreasing our objective function. That said, EM is less greedy in its approach than our proposed algorithm, which, in principle, seems like a good attribute. In practice, we doubt there would be any significant difference in the performance of these two approaches.

3. EXPERIMENTS AND RESULTS

Table 1 describes the collection of scores used in our evaluation, all obtained from the IMSLP website [1], consisting of 10 scores from 6 different publishers. In our evaluation we used about 20 pages from each score.

score index	IMSLP ID	pages	Publisher
1	24831	50-69	New York: Charles Foley
2	03631	2-21	Moscow: Muzgiz/Muzyka
3	65460	2-18, 21,22,23	
4	00569	2-19	
5	31875	2-21	Leipzig: Breitkopf & Härtel
6	01086	2-20	
7	06307	2-21	Leipzig: Ernst Eulenburg
8	00191	2-21	
9	08535	2-21	Vienna: Universal Edition
10	07354	2-25	Berlin: Schlesinger

Table 1. Information about the scores.

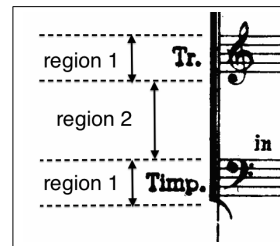


Figure 3. Two kinds of regions for possible text positions.

Our OMR system begins by computing the locations of the staves and the partition of staves into systems for each page of the score. These systems constitute the input to our system.

In all cases we use a graphical model based on the first page of the score, allowing for any subsequence of the named instruments, as in Figure 2. Several scores allow for the collective labeling of the strings with a single text tag, rather than an enumeration of instruments. For each $g \in G$ (i.e. each instrument) we supply the appropriate patterns, $P(g) = \{p_1, \dots, p_e\}$, by hand. Some instruments have two patterns, though most have only one. For all of the instruments and scores in our test set we only consider patterns where $p^l = 0$ meaning that the text must lie in the middle of the collection of staves associated with an instrument. Referring to Figure 3, this means we search in the text in region 1 when $p^k = 1$, and region 2 when $p^k = 2$, with obvious extensions to larger staff groupings. All of our test scores place instrument names in the left margin, though our approach easily accommodates other possible positions. We also supply the text strings, p^g , and the number of staves for each pattern, p^k . Even with instruments having two patterns, both used the same text (p^g) in our models.

score index	number of staves	number of errors			
		θ_0	θ_1	θ_2	θ_3
1	399	19	4	2	1
2	395	13	0		
3	317	21	10	4	0
4	331	44	0		
5	391	11	8	1	
6	443	33	2	0	
7	273	3	2	1	0
8	364	34	0		
9	381	4	0		
10	436	72	52	45	38
		θ_4	θ_5	θ_6	θ_7
		25	23	22	21

Table 2. Total number of staves and number of errors in each iteration for each score.

3.1 ORIGINAL TEMPLATES

The character template set includes all the (case sensitive) letters used in the instrument names of the 10 scores, in addition to a comma, hyphen, period, and space, giving 34 characters in all. Each score uses a subset of this collection in labeling instrument names. We create our original set of templates, θ_0 , by, for each character, randomly choosing an example from one of the 10 scores and thresholding the grey levels to choose probability models for each pixel. This collection will be the initial configuration for all 10 scores before we begin the adaptation process. We sampled from the test scores as a way of ensuring we could find examples of all the required templates, and note that, on average, only a tenth of these initial templates come from any individual score. A better scheme might use an omnifont model as our starting place.

We simultaneously estimate the staff labelings and the trained collection of character templates, iterating the approach of Section 2.4 until the results converge. Table 2 lists the total number of staves assigned incorrect instrument labels after each iteration of the algorithm, for each of the 10 scores. As shown in the table, 7 scores correctly labeled all the staves, 2 scores have only one labeling error, while the 10th score has 21 out of 436 staff labeling errors (4.82%), which is still low. The algorithm converged on all scores within 4 iterations, except for the the score containing the most errors, which used 8 iterations, as shown in the table.

As the original set of templates, θ_0 , comes from a variety of different scores with different fonts and sizes, they don't match any particular score font well. Our hope is that, through our iterative training process, the templates will *adapt* to the current score. Figure 4 gives an example from Score 6 with the original and learned characters drawn on top of the score image at their estimated locations. Clearly the original templates matched the actual font poorly, especially in size, as can be seen in the middle panel of the figure. The right panel of the figure shows the analogous result after two iterations of recognition and

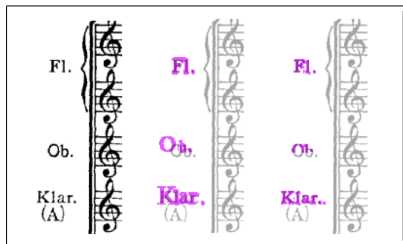


Figure 4. Comparing templates before (middle) and after (right) training. The instrument names in the actual score (left) are “Fl.,” “Ob.,” and “Klar.,” in order.

retraining. After this process most of the character templates match the font better, though not all of them. Due to the greedy nature of our algorithm it is necessary that our original templates, hence the original recognition and match, are close enough to pull the result into the correct local optimum. In this case the “O” and “b” in “Ob.” were misspecified and consistently matched poorly in the first iteration. As these characters don’t appear in other instrument names, there was no counteracting force helping to guide the models toward reasonable results, thus the outcome of the figure.

Although some of the trained results don’t look particularly good, Table 2 shows striking improvement in staff labeling due to training, *showing monotonic decrease in the number of errors*. Here is where the strength of the graphical model comes into play. Even with the poorly specified character models for “Ob.,” this is the only instrument name that can come between “Fl.” and “Klar.,” for which we have good models. This leads to the correct labeling in spite of the uneven training.

There are three scores having errors in our experiments. The one error in Score 1 is caused by unrelated text appearing in the left margin which was recognized as an incorrect instrument name. The one error in Score 5 mistakes “Vc.” as “Vla.” with “l” and “a” squeezed together. This happens in a three-staff system, thus the constraints imposed by graph ordering are less potent.

For Score 10, the errors are caused by badly trained templates. 8 out of 19 templates used in this score converged into unrecognizable glyphs. We suppose this happens because the font size of this score is obviously smaller than other scores and thus harder to adapt to. But surprisingly, this score still has reasonably accurate instrument labeling, which is our objective.

3.2 DROPPING THE GRAPHICAL CONSTRAINT

For comparison we ran a similar experiment without the ordering constraint imposed by the graph, thus allowing any group of staves to be labeled with any instrument. In this case all instrument orders are possible, even allowing for repetition of instruments. The results are shown in Table 3. After four iterations, the number of errors doesn’t seem to

score index	number of staves	number of errors			
		θ_0	θ_1	θ_2	θ_3
5	391	78	71	69	73
6	443	59	49	48	57
8	364	102	91	103	103

Table 3. Total number of staves and number of errors in each iteration for 3 scores without the graphical ordering constraint.

decrease. This is because many of the trained templates become unrecognizable — some of them become pure white space! Without the strong ordering constraint there is less gravitational pull toward the desired optimum, while we imagine the joint space of interpretations and models to be filled with local optima.

3.3 WORD MODELS VS. CHARACTER MODELS

Out of curiosity, we modified our model to view the instrument names as single rigid glyphs rather than character based models that allow for some flexibility in the placement of individual characters. Our experiments (not presented here) show that this approach works well when the actual document is consistent with the assumption we are making, but fails badly otherwise. Given the wide variety of typographical conventions encountered in music scores, we don't recommend this approach.

4. CONCLUSIONS

We have presented a method of interpreting the instrument name labels, which are a common way of labeling staves in large ensemble scores, showing nearly perfect recognition in all but one of the test scores we examined. The unusual aspect of our approach is that we simultaneously estimate both the labels we seek, as well as the text font used for the score. The experiments show convincing evidence that the strong assumption we make regarding possible labelings is powerful in practice and largely responsible for the reliability of the approach. In future work we will consider initial text models trained from a large variety of scores, as well as feature based, rather than template based, data models.

5. REFERENCES

- [1] IMSLP website. <http://imslp.org>.
- [2] Henry S Baird and George Nagy. Self-correcting 100-font classifier. In *IS&T/SPIE 1994 International Symposium on Electronic Imaging: Science and Technology*, pages 106–115. International Society for Optics and Photonics, 1994.
- [3] Issam Bazzi, Richard Schwartz, and John Makhoul. An omnifont open-vocabulary OCR system for english and arabic. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(6):495–504, 1999.
- [4] Jeff A Bilmes et al. A gentle tutorial of the EM algorithm and its application to parameter estimation for gaussian mixture and hidden markov models. *International Computer Science Institute*, 4(510):126, 1998.
- [5] Gary E Kopec and Mauricio Lomelin. Document image decoding approach to character template estimation. In *Image Processing, 1996. Proceedings., International Conference on*, volume 1, pages 213–216. IEEE, 1996.
- [6] Gary E Kopec and Mauricio Lomelin. Document-specific character template estimation. In *Electronic Imaging: Science & Technology*, pages 14–26. International Society for Optics and Photonics, 1996.
- [7] Ayatullah Faruk Mollah, Nabamita Majumder, Subhadip Basu, and Mita Nasipuri. Design of an optical character recognition system for camera-based handheld devices. *CoRR*, abs/1109.3317, 2011.
- [8] Shunji Mori, Ching Y Suen, and Kazuhiko Yamamoto. Historical review of OCR research and development. *Proceedings of the IEEE*, 80(7):1029–1058, 1992.
- [9] Christopher Raphael and Jingya Wang. New approaches to optical music recognition. In *ISMIR*, pages 305–310, 2011.
- [10] G. Read. *Music Notation: A Manual of Modern Practice*.
- [11] Zheng Song, Qiang Chen, Zhongyang Huang, Yang Hua, and Shuicheng Yan. Contextualizing object detection and classification. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1585–1592. IEEE, 2011.
- [12] Verena Thomas, Christian Wagner, and Michael Clausen. OCR based post processing of OMR for the recovery of transposing instruments in complex orchestral scores. In *Proceedings of the 12th International Society for Music Information Retrieval*, pages 411–416, 2011.
- [13] Frank Wessel and Hermann Ney. Unsupervised training of acoustic models for large vocabulary continuous speech recognition. *Speech and Audio Processing, IEEE Transactions on*, 13(1):23–31, 2005.

CROSS-VERSION SINGING VOICE DETECTION IN CLASSICAL OPERA RECORDINGS

Christian Dittmar¹ Bernhard Lehner² Thomas Prätzlich¹
Meinard Müller¹ Gerhard Widmer²

¹ International Audio Laboratories, Erlangen, Germany

² Johannes Kepler University, Linz, Austria

christian.dittmar@audiolabs-erlangen.de, bernhard.lehner@jku.at

ABSTRACT

In the field of Music Information Retrieval (MIR), the automated detection of the singing voice within a given music recording constitutes a challenging and important research problem. In this study, our goal is to find those segments within a classical opera recording, where one or several singers are active. As our main contributions, we first propose a novel audio feature that extends a state-of-the-art feature set that has previously been applied to singing voice detection in popular music recordings. Second, we describe a simple bootstrapping procedure that helps to improve the results in the case that the test data is not reflected well by the training data. Third, we show that a cross-version approach can help to stabilize the results even further.

1 Introduction

In classical opera, singing voice is considered to be one of the most important musical aspects. Locating vocal segments in an opera recording is an important prerequisite for applications such as singing voice separation or music structure analysis. The task of singing voice detection (also known as vocal detection) comprises automatic segmentation of a music recording into vocal (one or more singers) and non-vocal (accompaniment or silence) parts. A typical example of such a temporal segmentation is shown in Figure 1, where the black rectangles below each plot are ground truth segments and the red rectangles show automatically detected segments. The main challenge in automatic vocal detection comes both from the huge variety of singing voice characteristics as well as the simultaneous presence of other pitched musical instruments in the accompaniment. Especially in opera, the singers are often accompanied by instruments playing the same sequence of notes. Since the singers voice should dominate over the accompaniment, expressive techniques

such as pronounced vibrato and the so called singer’s formant [18] are often used. Moreover, the pitch and dynamic range of professional opera singers goes well beyond singing voices in popular music.

There has been quite some research on the problem of singing voice detection. The majority of previous contributions employ some sort of machine learning approach in combination with the extraction of audio features (see Section 2). When using machine learning, two major aspects need to be considered. First, appropriate audio features have to be designed that are suitable for the singing voice detection task. A delicate trade-off between elaborate, but error-prone extraction steps on the one hand, and undirected low-level features on the other hand has to be made. In this context, we introduce a novel extension to a previously proposed feature set and show that it is appropriate for singing voice detection.

Second, a supervised machine-learning algorithm usually learns from training data. It is well known that the performance of an optimized classifier can drop significantly if the “closed world” of the training data does not match the “open world” of the target data. A typical example is found in speech processing where systems trained with clean speech usually fail under noisy or reverberant conditions. One possibility to approach this challenge is so-called bootstrapping [14, 19]. As a second main contribution, we show how bootstrapping can help to improve singing voice detection by adapting classifiers to the specific recording under analysis. Furthermore, we describe a cross-version fusion approach [8] that can improve the results in case several versions of a music piece are available, which is a realistic assumption for opera and classical music in general.

2 Related Work

Although singing voice detection seems to be a task that is not so hard for human listeners, automatic singing voice detection remains difficult due to expressive characteristics of the singing voice and the diversity of accompaniment music playing simultaneously. These specific challenges have already been brought up in early works on the topic [2]. Given an unknown music recording, automatic singing voice detection is usually performed as a frame-wise estimation of singing voice activity. Even though this



© Christian Dittmar, Bernhard Lehner, Thomas Prätzlich, Meinard Müller, Gerhard Widmer. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Christian Dittmar, Bernhard Lehner, Thomas Prätzlich, Meinard Müller, Gerhard Widmer. “Cross-Version Singing Voice Detection in Classical Opera Recordings”, 16th International Society for Music Information Retrieval Conference, 2015.

poses a binary classification problem with just two classes, the acoustical variance within each class is so large that it is necessary to train the classifier with a wide range of training data.

Bootstrapping, i.e., the idea of using training data taken from the target recording itself, was proposed before as unsupervised [14] and user assisted [19] strategy for improving classification performance. One of the first attempts to separate the singing voice from the accompaniment prior to the feature extraction stage was described in [20]. Post-processing of the so-called posterior probabilities obtained during classification was described in [12].

A large set of low-level features was used in conjunction with a Support Vector Machine (SVM) classifier in [15]. Furthermore, the authors published singing voice annotations for training, validation and test subsets of the JAMENDO corpus, enabling reproducible comparisons between different methods (see Section 5.2). The same test corpus was used for evaluation in [16], where the feature extraction focused on vibrato and tremolo properties. A study on the effect of accompaniment music in singing vs. rap discrimination was presented in [6]. Very promising results in singing voice detection and related tasks were reported in [13]. However, the proposed signal processing chain was quite elaborate and involved an estimation of the predominant pitch, which can lead to substantial error propagation to all the feature extractors depending on it.

Lehner et al. [10] focused on achieving comparable results using a light-weight approach. In a follow-up work, they improved the achievable precision by introducing novel audio features tailored to the singing voice detection scenario [11]. A recent paper [4] showed that two cross-version post-processing strategies can improve the singing voice detection performance achievable with the light-weight feature set of [10, 11].

So far, the best classification performance on the JAMENDO data set was reported in [9], using a Bidirectional Long Short-Term Memory Recurrent Neural Network as machine learning scheme that inherently takes the temporal context of low-level feature sequences into account. However, it reads as if the authors selected the optimal network architecture according to the best results obtained w.r.t. the test set instead of the validation set. Thus, we think that their results might be overly optimistic.

3 Baseline Singing Voice Detection

Our baseline system for singing voice detection closely follows the approach proposed in [10, 11]. The extraction of descriptive audio features is performed by splitting the audio signals into frames and transforming each frame to the spectral domain. Low-level and mid-level audio features are computed from each resulting spectral frame, forming a feature vector by concatenation. Supervised machine learning is employed to train a classifier for discriminating the feature vector assigned to each frame into the two classes vocal and non-vocal. Note that the vocal class usu-

Feature name and reference	Abbrev.	Dim.
Mel-frequency Cepstral Coefficients [10]	MFCC	30
Vocal Variance [11]	VOCVAR	5
Fluctogram Variance [11]	FLUCT	17
Spectral Contraction Variance [11]	NSD	17
Spectral Flatness Mean [11]	FLAT	17
Polynomial Shape Spectral Contrast [1,7]	PSSC	24

Table 1. Feature names, abbreviations, and dimensionality of the low-level and mid-level audio features used.

ally comprises singing voice plus accompaniment, which makes the task more intricate.

3.1 Feature Extraction and Processing

Table 1 lists the complete set of features that is used in our approach. Since most of our descriptors are wellknown in the MIR literature, we only highlight a few aspects here.

Mel-Frequency Cepstral Coefficients (MFCC) are one of the most common audio features widely used in diverse audio classification tasks. They are designed to capture the spectral envelope of an audio signal using only a few coefficients in the so-called Cepstral domain. As described in [10], we use an optimized parametrization with a different time-frequency resolution and a higher number of coefficients than usual. A strongly related feature is the **Vocal Variance**, which basically captures the variance in the first 5 MFCCs across a number of consecutive frames.

The mid-level features **Fluctogram**, **Spectral Contraction**, and **Spectral Flatness** are the most important contributions from [11]. All three are extracted in 17 overlapping frequency bands, where each band covers two octaves and neighboring bands are spaced three semitones apart. The Fluctogram encodes the relative frequency fluctuation of salient tonal components in each band, without the need for an actual estimation of a predominant pitch. Spectral Contraction and Flatness are designed to complement the Fluctogram, encoding whether there are reliable harmonic components with clear sinusoidal peaks or rather a noise-like distribution of the spectrum within the current band boundaries.

Spectral Contrast encodes the relation of peaks to valleys of the spectral magnitude in several sub bands. The band boundaries have been specified for the Octave-Based Spectral Contrast (OBSC) [7] and the Shape-Based Spectral Contrast (SBSC) [1]. In general, both variants can be interpreted as harmonicity or tonality descriptor. We suggest a modification of the already existing methods, both of which were successfully used for music genre classification tasks. In the previous approaches, the spectral magnitude values in each sub band are sorted and the relation between the lowest and highest fraction is encoded via statistical measures. In our modification, we propose to fit a third-order polynomial to the ordered magnitude values and store the three polynomial coefficients together with the offset as descriptors. Therefore, we refer to this feature as **Polynomial Shape Spectral Contrast (PSSC)**. It is

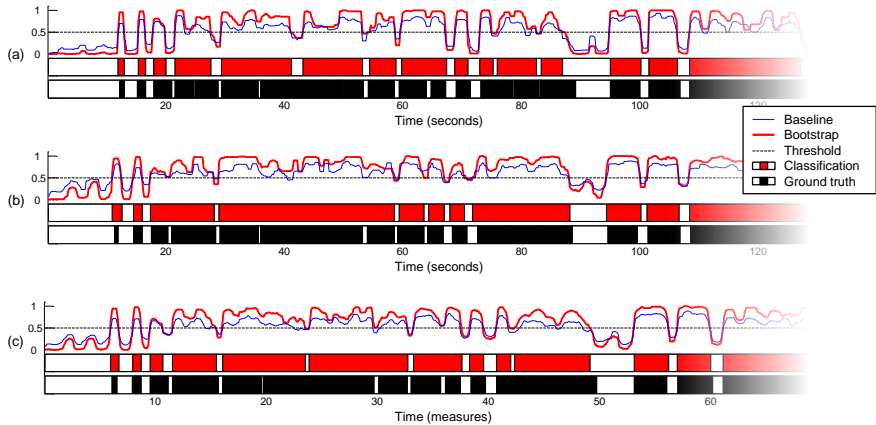


Figure 1. Illustration of the cross-version post-processing strategies as described in Section 4.1 and Section 4.2. The curves and annotations are based on an excerpt corresponding to the first 80 measures of the duet No. 6 (Agathe and Ännchen): “Schelm! halt fest” from the opera “Der Freischütz” by Carl Maria von Weber. For each case, the decision functions of the baseline (blue thin curve) and bootstrap (red bold curve) classifier are shown. The colored time-lines below the decision curves show the automatically detected singing voice activity (red segments, derived from bootstrap decision) vs. the ground truth (black segments). (a): Recording of the performance conducted by Karl-Heinz Bloemecke (2013). (b): Recording of the performance conducted by Carlos Kleiber (1973). (c): Cross-version results based on three performances (including Bloemecke and Kleiber) after temporal alignment to a common, measure-based time axis and subsequent averaging across the individual decision functions.

computed for each of the 6 sub bands (0-200 Hz, 200-400 Hz, 400-800 Hz, 800-1600 Hz, 1600-3200 Hz, and 3200-8000 Hz), yielding a feature vector with 24 attributes. In contrast to the procedure in [1, 7], we do not apply any decorrelation procedure to the raw features, hence reducing the computational complexity. Compared to the before mentioned versions of spectral contrast, our modification resulted in better accuracy on our internal data set (PSSC: 80.2%, OBSC: 73.4%, and SBSC: 72.3%). In total, the concatenation of all features listed in Table 1 results into a 110-dimensional feature vector per spectral frame. The set of all feature vectors makes up our feature matrix which is split into appropriate training and test sets and used for machine learning in the following.

3.2 Classification and Decision Function

Again following [10, 11], we employ Random Forests (RF) [3] as classification scheme. RF are an instance of the so-called Bootstrap Aggregation (Bagging) concept applied to Classification and Decision Trees (CART) [21] classifiers. This machine learning ensemble meta algorithm was designed to improve the stability and accuracy by averaging over a set of weak classifiers trained from random subspaces of the complete feature matrix. In RF, random sets of CARTs are trained by introducing randomness at 2 levels: in the subset of features as well as in the subset of

training data [3]. The generalization error of RF depends on the classification strength of the individual CARTs as well as their mutual correlation. As changes in the feature selection cause drastic changes in the tree structure, the individual trees are expected to be uncorrelated. Averaging their individual decisions in the RF leads to decreased variance of the classifier model, which is in general a desirable property.

RFs deliver a frame-wise score value per class that can be interpreted as confidence measure for the classifier decision. In our binary classification scenario, the two score functions are inversely proportional. We pick the one corresponding to our target vocal class and refer to it as decision function in the following. A decision function value close to 1 indicates a very reliable assignment to the vocal class, whereas a value close to 0 points to the non-vocal class. In order to binarize the decision function, we compare it to a threshold. Only frames where the decision function value exceeds the threshold will be classified as vocal. Prior to that, the decision function is smoothed using a median filter. The filter width given in seconds is an important parameter. Median filtering of the decision function is justified by the observation that singing voice activity usually exhibits a certain continuity. So this step helps to stabilize the detection result and to prevent unreasonably short gaps in the decision function, where the classification rapidly flips from vocal to non-vocal or vice versa.

4 Post-processing of Singing Voice Detection

In this section, we describe two approaches suitable for post-processing of intermediate singing voice detection results. First, we describe our approach to unsupervised bootstrap training of a classifier adapted to the recording under analysis. Second, we describe how to perform a late fusion of decision functions by means of time alignment between different versions.

4.1 Bootstrap Training

Inspired by the ideas in [14, 19], we propose to perform a second, specialized RF classification subsequent to the initial singing voice detection stage. The rationale is to remedy the “closed world” vs. “open world” training problem discussed before (see Section 1). We do so by creating an adapted classifier model that is trained with feature vectors exclusively taken from the current recording under analysis. However, this recording does usually not come together with an annotation of its frames to the two classes. So how to assign the feature vectors automatically to the training sets of the vocal respective non-vocal class?

Our idea is to base this assignment on the shape of the decision function generated by the initial RF classifier. Looking at the course of this decision function, we see some extreme values for frames, where the observed feature vectors match very well to either the vocal or non-vocal class of the initial classifier model. However, many values reside in the middle of the range of decision function values, where an assignment to either side is questionable. If we now select two subsets of the feature vectors, each corresponding to an upper and lower fraction (e.g., 20%) of the range of decision function values, we can use these to train a small RF classifier that is adapted to the feature space spanned by the recording under analysis. Before we do so, we stratify the training set, meaning that we randomly select the same number of feature vectors for each class from the subset corresponding to the upper and lower decision values.

In Figure 1, we observe that the new decision functions (red curve) generated by classifying the current song with the adapted RF classifier exhibits a more desirable shape than the decision function generated by the initial RF classifier (blue curve). In Figure 1(a), it can be seen, that the bootstrap decision function can close small gaps, where the initial decision function dipped below the decision threshold (e.g., at around 80 s).

4.2 Cross-Version Fusion

In [8], Konz et al. introduced the intuitive yet effective idea to exploit the availability of different recordings of the same piece of music for stabilizing automatic chord recognition results. We pursue the same idea here in order to perform a late fusion of decision functions obtained from the initial singing voice detection. This is achieved by

Authors and Reference	Accuracy	F-measure
Biased Guess (all frames vocal)	46.3	0.64
Vembu and Baumann 2005 [20]	77.4	0.77
Ramona et al. 2008 [15]	82.2	0.84
Regnier and Peeters 2009 [16]	—	0.77
Lehner et al. 2013 [10]	84.8	0.85
Lehner et al. 2014 [11]	88.2	0.87
Leglaive et al. 2015 [9]	91.5	0.91
Proposed feature set	88.2	0.87

Table 2. Singing voice detection results achievable with our novel feature set in comparison to other authors. The basis of all measurements is a publicly available subset of the JAMENDO corpus [15].

warping the individual decision functions obtained for different versions of the same piece to a version-independent representation with a musical time axis given in measures (respective sub-divisions thereof) instead of seconds. For the moment, we assume that the required temporal position of measure boundaries is given. In Section 5.3, we sketch how to retrieve the measure boundaries automatically.

In general, the procedure described above yields a set of time-aligned decision functions that we use to derive a fused, overall decision function. To this end, we use the most straightforward approach and just take the arithmetic mean of the decision values of all aligned decision functions. The averaging is intended to compensate for noise and artifacts that might occur in the individual decision functions. Figure 1(c) presents the resulting decision function on the measure-related time axis. We show the fused decision function derived from baseline singing voice detection (thin blue curve) overlaid with the fused decision function derived from bootstrap training (bold red curve). It can be seen that the averaging leads to a slightly more stable decision function. Comparison of the fused bootstrap decision function against the decision threshold (dashed black line) yields our estimated singing voice segments (black rectangles). In general, the estimated segments exhibit improved agreement to the ground truth segmentation in comparison to Figure 1(a) and 1(b).

5 Evaluation

In this section, we assess the performance of our proposed methods. First, we validate our novel feature set on a public benchmark data set. Second, we show that bootstrapping and cross-version fusion can help to improve the results for classical opera recordings.

5.1 Experimental Settings

For our experiments, we are going to fix the following parameters: For the majority of features in Table 1, the hopsize between consecutive analysis frames is 200 ms (fea-

ture rate of 5 Hz), the analysis windows have a length of 800 ms. The raw flucrogram, flatness and contraction features are extracted on a finer temporal level, with a hop-size of 20 ms and a window size of 100 ms. We aggregate 40 consecutive frames of these raw features and use their variance as descriptor for flucrogram and contraction, and their means as descriptor for flatness. In the RF classifier, we use 128 individual CART classifiers, each trained with a randomly selected subset of 5 feature dimensions, from the originally 110-dimensional feature space. For post-processing of the decision functions, we employ a median filter with a width of 1.4 s. The decision function threshold is set to 0.5. In the next sections, we keep these settings fixed for the evaluation of our baseline system as well as our proposed post-processing strategies.

5.2 Performance on a Common Benchmark

In order to benchmark our novel feature set against the state-of-the-art, we used a subset of the publicly available JAMENDO music corpus [15]. Each recording in that data set was manually annotated into vocal and non-vocal sections by the original author. Since human annotators can have difficulties in determining singing segment boundaries, the segmentation allowed some uncertainty, i.e., very short instrumental breaks were not labeled as such. The exact split into training, validation and test set is specified in [15]. Table 2 lists our results in comparison to previously published works. The used metrics are the frame-wise F-measure and the accuracy which are computed by evaluating all frames across the 16 test songs. According to the ground truth annotation, the majority of frames belongs to the non-vocal class. We also report the **Biased Guess**, where all frames of a test item are assigned to the vocal class, because in classical opera, the vocal class usually occurs more often. As can be seen, the performance of our proposed feature set is on par with the state-of-the-art. Only the accuracy and F-measure reported in [9] surpass our results, but the comparison might not be entirely fair as discussed in Section 2.

5.3 Opera Case-Study

The opera “Der Freischütz” by Carl Maria von Weber, a work of high relevance for opera studies, was chosen for the further evaluation. For this opera, there exists a large number of historical sources, including a multitude of audio recordings. In the project “Der Freischütz Digital”¹, musicologists and computer scientists cooperate to explore opportunities for new and digital ways of research, analysis and presentation of music related data in critical editions [17].

From the corpus used in the project, we had three different versions of this opera available for the purpose of cross-version singing voice detection. The respective conductors

Opera	Conductor	Year
“Carmen”	Lorin Maazel	1984
“Die Zauberflöte”	Nikolaus Harnoncourt	1988
“Pelleas et Melisande”	Claudio Abbado	1992
“La Cenerentola”	Riccardo Chailly	1993
“La Traviata”	Carlo Rizzi	2005
“Tristan und Isolde”	Daniel Barenboim	1995
“Der Freischütz”	Karl Elmendorff	1944
“Der Freischütz”	Carlos Kleiber	1973
“Der Freischütz”	Karl-Heinz Bloemcke	2013

Table 3. Overview over the used opera recordings. The upper half specifies the operas available as training set, the lower half gives the operas used as test set.

and recording years are shown in Table 3. All numbers in the three versions have orchestral accompaniment and varying number of soloist singers. We picked the numbers 6, 8, and 9 as test cases of different musical complexity, a duet, a solo aria and a trio, respectively.

For evaluation purposes, we first had to generate reference annotations of the singing voice activity in these pieces. This was achieved semi-automatically by means of aligning a MIDI version of each piece to the recording and taking the note onsets and offsets of the singing voice as reference. Details about this procedure can be found in [5].

Furthermore, each recording had its measures (i.e., the beginning of each bar) manually annotated to facilitate the alignment between corresponding versions of the same number. The manually annotated bar positions are used to warp the individual decision functions to a common time axis regardless of their original tempo and variations thereof.

5.4 Results and Discussion

The diagrams in Figure 2 illustrate the benefit of applying bootstrap training (see Section 4.1), cross-version fusion (see Section 4.2), as well as a combination of both in two different training scenarios. The bar plots in both (a) and (b) show the F-measures obtained per test item as well as the average F-measure value. The following singing voice detection and post-processing strategies were tested. **Random Guess** refers to randomly assigning the frames of our test data to either the vocal or non-vocal class with equal probability. Since the vocal class occurs more frequently in our test data, the resulting F-Measure is slightly above chance. **Biased Guess** refers to assigning the singing voice class to each frame of a test recording. It can be seen that the resulting F-measure is already quite high, again a consequence of the dominance of the vocal class in our test set. **Baseline Detection** refers to the results obtained by the baseline singing voice detection system as described in Section 3. **Bootstrap Detection** refers to the results obtained by a second classification run with an adapted RF classifier using the bootstrapping strategy as described in Section 4.1. **Cross-version Fusion** refers to the results of

¹ www.freischuetz-digital.de

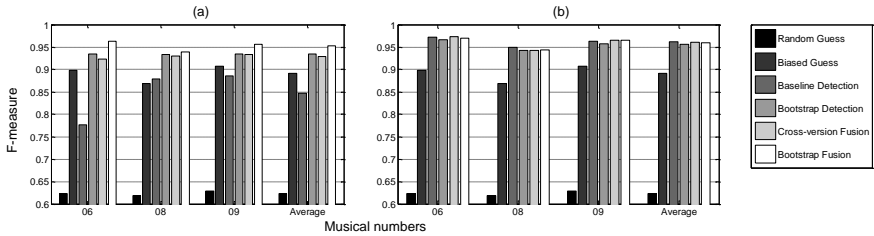


Figure 2. The average F-measures obtained in two different training scenarios and four post-processing strategies. The test set consisted of three versions of the numbers 6, 8, and 9 from the opera “Der Freischütz.” (a): Results obtained by training the initial RF classifier with popular music recordings from the RWC and JAMENDO data sets. (b): Results obtained by training the initial RF classifier with classical opera recordings not including “Der Freischütz.”

fusing the initial RF decision functions of all available versions of each test recording as described in Section 4.2. Finally, **Bootstrap Fusion** refers to the results obtained by combining both the bootstrap training and the cross-version fusion.

The results in Figure 2(a) were obtained by training the initial RF classifier with a combined data set comprising both the JAMENDO [15] and RWC [13] subsets that are annotated for singing voice. Both corpora are dominated by recordings of popular music. Obviously, this kind of training material differs from the music content in the test set. The average singing voice detection performance stays even below the biased guess. However, this rather poor initial estimate for the vocal frames can be used for bootstrap training. Consequently, the bootstrap training leads to a substantial performance gain, surpassing the bias results. Cross-version fusion of the imperfect initial decision functions leads to similar improvements as the bootstrap training. The combination of both bootstrap training and cross-version fusion of decision functions delivers the best results in this training scenario.

The results in Figure 2(b) were obtained when training the initial RF classifier with recordings of classical opera. Specifically, we used the operas listed in the upper half of Table 3. In total, the playtime of our training material amounts to approximately 4 h. As can be seen from the F-measure of the baseline RF classifier, this kind of training data gives a considerable performance boost. This is not surprising, since the orchestral timbre as well as the pronounced use of vibrato singing in these opera recordings is very similar to our test items. The remaining F-measures show that the proposed post-processing strategies at best lead to marginal improvements since the performance is already saturated.

From our comparison, we infer that bootstrap training could be recommended as standard post-processing strategy for singing voice detection in classical opera recordings. This is especially true if the initial classification delivers reasonable results that can be surpassed if more appropriate training data would be available. However, bootstrap training does not seem to help much if there exists no

combination of feature set, training set, and classifier that can obtain good singing voice detection for the recording under analysis. Moreover, bootstrap training has the drawback that it will likely produce erroneous decision functions when there is no singing voice activity at all throughout a recording. If these cases can not be ruled out from bootstrap training, singing voice detection results could even deteriorate in comparison to the baseline system.

6 Conclusions and Future Work

In this paper, we made two contributions to advancing the state-of-the-art in automatic singing voice detection. First, we proposed a novel extension to a state-of-the-art audio feature set for singing voice detection and validated it on a public benchmark set. Second, we proposed bootstrap training and cross-version fusion as post-processing strategies applicable to intermediate results from a machine learning system. In our case study, involving multiple recordings of Carl Maria von Webers opera “Der Freischütz,” we have shown that a combination of bootstrap training and cross-version fusion can help to improve the classification performance if the training data is very different from the test data. While bootstrap fusion might be applicable to improve singing voice detection in various music genres, cross-version fusion can only help if we have multiple, sufficiently similar versions of the same piece of music available. Future work will be directed towards further refinements and applications of these techniques for various kinds of music genres.

7 Acknowledgments

This work has been supported by the BMBF project Freischütz Digital (Funding Code 01UG1239A to C), and by the Austrian Science Fund (FWF) under grants TRP307-N23 and Z159. The International Audio Laboratories Erlangen (AudioLabs) is a joint institution of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) and Fraunhofer IIS.

8 References

- [1] Vincent Akkermans and Joan Serrá. Shape-based spectral contrast descriptor. In *Proc. of the Sound and Music Computing Conf. (SMC)*, pages 143–148, Porto, Portugal, July 2009.
- [2] Adam L. Berenzweig and Daniel P. W. Ellis. Locating singing voice segments within music signals. In *Proc. of the IEEE Workshop on the Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 119–122, New Paltz, New York, USA, October 2001.
- [3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.
- [4] Christian Dittmar, Thomas Prätzlich, and Meinard Müller. Towards cross-version singing voice detection. In *Proc. of the Jahrestagung für Akustik (DAGA)*, Nuremberg, Germany, March 2015.
- [5] Sebastian Ewert, Meinard Müller, and Peter Grosche. High resolution audio synchronization using chroma onset features. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1869–1872, Taipei, Taiwan, April 2009.
- [6] Daniel Gärtner and Christian Dittmar. Vocal characteristics classification of audio segments: An investigation of the influence of accompaniment music on low-level features. In *Proc. of the Int. Conf. on Machine Learning and Applications (ICMLA)*, pages 583–589, Miami, Florida, USA, December 2009.
- [7] Daning Jiang, Lie Lu, Hong-Jiang Zhang, Jian-Hua Tao, and Lian-Hong Cai. Music type classification by spectral contrast feature. In *Proc. of the IEEE Int. Conf. on Multimedia and Expo (ICME)*, volume 1, pages 113–116, Lausanne, Switzerland, August 2002.
- [8] Verena Konz, Meinard Müller, and Rainer Kleinertz. A cross-version chord labelling approach for exploring harmonic structures—a case study on Beethoven’s *Appassionata*. *Journal of New Music Research*, 42(1):1–17, January 2013.
- [9] Simon Leglaive, Romain Hennequin, and Roland Badeau. Singing voice detection with deep recurrent neural networks. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 121–125, Brisbane, Australia, April 2015.
- [10] Bernhard Lehner, Reinhard Sonnleitner, and Gerhard Widmer. Towards lightweight, real-time-capable singing voice detection. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pages 53–58, Curitiba, Brazil, November 2013.
- [11] Bernhard Lehner, Gerhard Widmer, and Reinhard Sonnleitner. On the reduction of false positives in singing voice detection. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 7480–7484, Florence, Italy, May 2014.
- [12] Hanna Lukashevich and Christian Dittmar. Effective singing voice detection in popular music using arma filtering. In *Proc. of the Int. Conf. on Digital Audio Effects (DAFx)*, pages 165–168, Bordeaux, France, September 2007.
- [13] Matthias Mauch, Hiromasa Fujihara, Kazuyoshii Yoshii, and Masataka Goto. Timbre and melody features for the recognition of vocal activity and instrumental solos in polyphonic music. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pages 233–238, Miami, Florida, USA, October 2011.
- [14] Tin Lay Nwe and Ye Wang. Automatic detection of vocal segments in popular songs. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pages 138–144, Barcelona, Spain, October 2004.
- [15] Mathieu Ramona, G ael Richard, and Bertrand David. Vocal detection in music with support vector machines. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, pages 1885–1888, Las Vegas, Nevada, USA, March 2008.
- [16] Lise Regnier and Geoffroy Peeters. Singing voice detection in music tracks using direct voice vibrato detection. In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1685–1688, Taipei, Taiwan, April 2009.
- [17] Daniel R owenstrunk, Thomas Pr atzlich, Thomas Betzwieser, Meinard M uller, Gerd Szwillus, and Joachim Veit. Das Gesamtkunstwerk Oper aus Datensicht - Aspekte des Umgangs mit einer heterogenen Datenlage im BMBF-Projekt "Freischutz Digital". *Datenbank-Spektrum*, 15(1):65–72, 2015.
- [18] Zheng Tang and Dawn A. A. Black. Melody extraction from polyphonic audio of western opera: A method based on detection of the singer’s formant. In *Proc. of the Int. Society for Music Information Retrieval Conf. (ISMIR)*, pages 161–166, Taipei, Taiwan, October 2014.
- [19] George Tzanetakis. Song-specific bootstrapping of singing voice structure. In *Proc. of the IEEE Int. Conf. on Multimedia and Expo (ICME)*, volume 3, pages 2027–2030, Taipei, Taiwan, June 2004.
- [20] Shankar Vembu and Stefan Baumann. Separation of vocals from polyphonic audio recordings. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, pages 337–344, London, UK, September 2005.
- [21] Kristopher West and Stephen Cox. Features and classifiers for the automatic classification of musical audio signals. In *Proc. of the Int. Conf. on Music Information Retrieval (ISMIR)*, Barcelona, Spain, October 2004.

Accurate Tempo Estimation based on Recurrent Neural Networks and Resonating Comb Filters

Sebastian Böck, Florian Krebs and Gerhard Widmer

Department of Computational Perception
Johannes Kepler University, Linz, Austria

sebastian.boeck@jku.at

ABSTRACT

In this paper we present a new tempo estimation algorithm which uses a bank of resonating comb filters to determine the dominant periodicity of a musical excerpt. Unlike existing (comb filter based) approaches, we do not use hand-crafted features derived from the audio signal, but rather let a recurrent neural network learn an intermediate beat-level representation of the signal and use this information as input to the comb filter bank. While most approaches apply complex post-processing to the output of the comb filter bank like tracking multiple time scales, processing different accent bands, modelling metrical relations, categorising the excerpts into slow/fast or any other advanced processing, we achieve state-of-the-art performance on nine of ten datasets by simply reporting the highest resonator's histogram peak.

1. INTRODUCTION

Tempo estimation is one of the most fundamental music information retrieval (MIR) tasks. The tempo of music corresponds to the frequency of the beats, i.e. the speed at which humans usually tap to the music.

In this paper, we only deal with global tempo estimation, i.e. report a single tempo estimate for a given musical piece, and do not consider the temporal evolution of tempo. Possible applications for such algorithms include automatic DJ mixing, similarity estimation, music recommendation, playlist generation, and tempo aware audio effects. Finding the correct tempo is also vital for many beat tracking algorithms which use a two-folded approach of first estimating the tempo of the music and then aligning the beats accordingly.

Many different methods for tempo estimation have been proposed in the past. While early approaches estimated the tempo based on discrete time events (e.g. MIDI notes or a sequence of onsets) [6], almost all of the recently proposed algorithms [4, 7, 8, 17, 23, 28] use some kind of continuous input. Generally, they follow this procedure: they trans-

form the audio signal into a down-sampled feature, estimate the periodicities and finally select one of the periodicities as tempo.

As a reduction function, the signal's envelope [26], band pass filters [8, 17, 28], onset detection functions [4, 8, 23, 28] or combinations thereof are commonly used. Popular choices for periodicity detection include Fast Fourier Transform (FFT) based methods like tempograms [3, 28], autocorrelation [6, 8, 23, 25] or comb filters [4, 17, 26]. Finally, post-processing is applied to chose the most promising periodicity as perceptual tempo estimate. These post-processing methods range from simply selecting the highest periodicity peak to more sophisticated (machine learning) techniques, e.g. hidden Markov models (HMM) [17], Gaussian mixture model (GMM) regression [24] or support vector machines (SVM) [9, 25].

In this paper, we propose to use a neural network to derive a reduction function which makes complex post-processing redundant. By simply selecting the comb filter with the highest summed output, we achieve state-of-the-art performance on nine of ten datasets in the *Accuracy 2* evaluation metric.

2. RELATED WORK

In the following, we briefly describe some important works in the field of tempo estimation. Gouyon et al. [12] give an overview of the first comparative algorithm evaluation which took place for ISMIR 2004, followed by another study by Zapata and Gómez [29].

The work of Scheirer [26] was the first one to process the audio signal continuously rather than working on a series of discrete time events. He proposed the use of resonating comb filters, which are one of the main techniques used for periodicity estimation since then. Periodicity analysis is performed on a number of band pass filtered signals and then the outputs of this analysis are combined and a global tempo is reported.

Dixon [6] uses discrete onsets gathered with the spectral flux method to build clusters of inter onset intervals which are in turn processed by a multiple agent system to find the most likely tempo. Oliveira et al. [23] extend this approach to use a continuous input signal instead of discrete time events and modified it to allow causal processing.

Klapuri et al. [17] jointly analyse the musical piece at three time scales: the tatum, tactus (which corresponds to



© Sebastian Böck, Florian Krebs and Gerhard Widmer. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Sebastian Böck, Florian Krebs and Gerhard Widmer. "Accurate Tempo Estimation based on Recurrent Neural Networks and Resonating Comb Filters", 16th International Society for Music Information Retrieval Conference, 2015.

the beat or tempo) and measure level. The signal is split into multiple bands and then combined into four accent bands before being fed into a bank of resonating comb filters similar to [26]. Their temporal evolution and the relation of the different time scales are modelled with a probabilistic framework to report the final position of the beats. The tempo is then calculated as the median of the beat intervals during the second half of the signal.

Instead of a multi-band approach as used in [17, 26], Davies and Plumley [4] process an autocorrelated version of a complex domain onset detection function with a shift invariant comb filter bank to get the beat period. Although this method uses only a single dimensional input feature, it performs almost as good as the competing algorithms in [12] but has much lower computational complexity.

Gainza and Coyle [8] use a multi-band decomposition to split the audio signal into three frequency bands and then perform a transient/onsets detection (with different onset detection methods). These are transformed via autocorrelation into periodicity density functions, combined, and weighted to extract the final tempo.

Gkiokas et al. [9] utilise harmonic/percussive source separation on top of a constant-Q transformed signal in order to extract chroma features and filter bank energies from the separated signal respectively. Periodicity is estimated for both representations with a bank of resonating comb filters for overlapping windows of 8 seconds length and the resulting features are combined before a metrical level analysis is performed to report the final tempo. In a consecutive work [10] they use a support vector machine (SVM) to classify the music into tempo classes to better predict the tempo to be reported.

Elowsson et al. [7] also use harmonic/percussive source separation to model the speed of music. They derive various features like onset densities (for multiple frequency ranges) and strong onset clusters and use a regression model to predict the tempo of the signal.

Percival and Tzanetakis [25] use a “traditional” approach by first generating a spectral flux onset strength signal, followed by a stage which detects the beat period in overlapping windows of approximately 6 seconds length (via generalised autocorrelation with harmonic enhancement) and a final accumulating stage which gathers all these tempo estimates and uses a support vector machine (SVM) to decide which octave the tempo should be in.

Wu and Jang [28] first derive an unaltered and a low pass filtered version of the input signal. Then they obtain a tempogram representation of a complex domain onset detection function for both signals to obtain tempo pairs. A classifier is then used to report the final most salient tempo.

3. ALGORITHM DESCRIPTION

Scheirer [26] found it beneficial to compute periodicities individually on multiple frequency bands and then subsequently combine them to estimate a single tempo. Klapuri et al. [17] followed this route but Davies and Plumley argued that is is enough to have a single – musically meaningful – feature to estimate the periodicity of a signal [4].

Given the fact that beats are the musically most relevant descriptors for the tempo of a musical piece, we take this approach one step further and do not use the pre-processed signal directly – or any representation that is strongly correlated with it, e.g. an onset detection function – as an input for a comb filter, but rather process the signal with a neural network which is trained to predict the positions of beats inside the signal. The resulting beat activation function is then fed into a bank of resonating comb filters to determine the tempo.

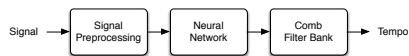


Figure 1: Overview of the new tempo estimation system.

Figure 1 gives general overview over the different steps of the tempo estimation system, which are described into more detail in the following sections.

3.1 Signal Pre-Processing

The proposed system processes the signal in a frame-wise manner. Therefore the audio signal is split into overlapping frames and weighted with a Hann window of same length before being transferred to a time-frequency representation by means of the Short-time Fourier Transform (STFT). Two adjacent frames are located 10 ms apart, which corresponds to a rate of 100 fps (frames per second). We omit the phase portion of the complex spectrogram and use only the magnitudes for further processing. To reduce the dimensionality of the signal, we process it with a logarithmically spaced filter which has three bands per octave and is limited to the frequency range [30, 17000] Hz. To better match the human’s perception of loudness, we scale the resulting frequency bands logarithmically. As the final input features for the neural network, we stack three spectrograms and their first order difference calculated with different STFT sizes of 1024, 2048 and 4096 samples, a visualisation is given Figure 2b.

3.2 Neural Network Processing

As a network we chose the system presented in [1], which is also the basis for the current state-of-the-art in beat tracking [2, 18]. The output of the neural network is a beat activation function, which represents the probability of a frame being a beat position. Instead of processing the beat activation function to extract the positions of the beats, we use it directly as a one-dimensional input to the bank of resonating comb filters.

Using this continuous function instead of discrete beats is advantageous since the detection is never 100% effective and thus introduces errors when inferring the tempo directly from the beats. This is in line with the observation that recent tempo induction algorithms use onset detection functions or other continuously valued inputs rather than discrete time events.

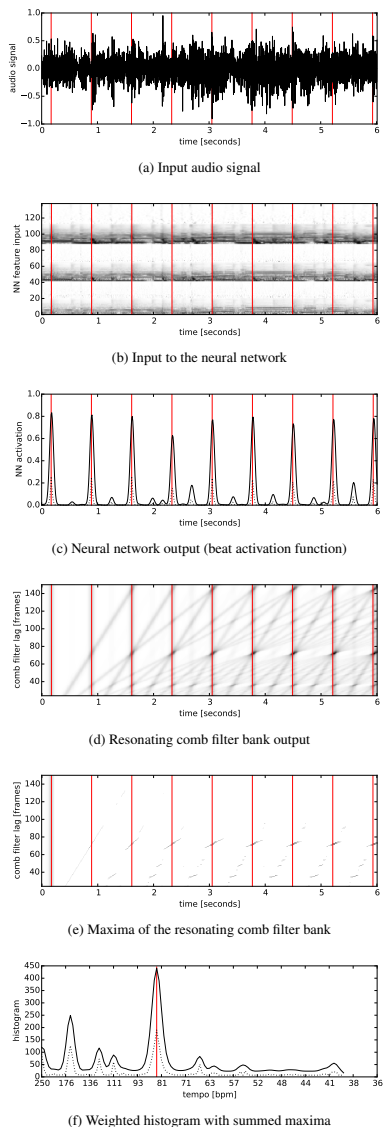


Figure 2: Signal flow of a 6 second pop song excerpt: (a) input audio signal, (b) pre-processed input to the neural network, (c) its raw (dotted) and smoothed (solid) output, (d) corresponding comb filter bank response, (e) the maxima thereof, (f) resulting raw (dotted) and smoothed (solid) weighted histogram of the summed maxima. The beat positions and the tempo are marked with vertical red lines.

We believe that the learned feature representation (at least to some extent) incorporates information that otherwise would have to be modelled explicitly, either by tracking multiple time scales [17], processing multiple accent bands [26], modelling metrical relations [9], dividing the excerpts into slow / fast categories [7] or any other advanced processing. Figure 2c shows an exemplary output of the neural network. It can be seen that the network activation function has strong regular peaks that do not always coincide with high energies in the network’s inputs.

3.2.1 Network Training

We train the network on the datasets described in Section 4.2 which are marked with an asterisk (*) in an 8-fold cross validation setting based on a random splitting of the datasets. We initialise the network weights and biases with a uniform random distribution with range $[-0.1, 0.1]$ and train it with stochastic gradient descent with a learning rate of 10^{-4} and a momentum of 0.9. We stop training if no improvement of the cross entropy error of the validation set can be observed for 20 epochs. All adjustable parameters of the system are tuned to maximise the tempo estimation performance on the validation set.

3.2.2 Activation Function Smoothing

The beat activation function of the neural network reflects the probability that a given frame is a beat position. However, it can happen that the network is not sure about the exact position of the beat if it falls close to the border between two frames and hence splits the reported probability between these two frames. Another aspect to be considered is the fact that the ground truth annotations used as targets for the training are sometimes generated via manual tapping and thus deviate from the real beat position by up to 50 ms. This can result also in blurred peaks in the beat activation function. To reduce the impact of these artefacts, we smooth the activation function before being processed with the filter bank by convolving it with a Hamming window of length 140 ms.¹

3.3 Comb Filter Periodicity Estimation

We use the output of the neural network stage as input to a bank of resonating comb filters. As outlined previously, comb filters are a common choice to detect periodicities in a signal, e.g. [4, 17, 26]. The advantage of comb filters over autocorrelation lays in the fact that comb filters also resonate at multiples, fractions and simple rationales of the filter lag. This behaviour is in line with the perception of humans, which do not necessarily consider double or half tempi wrong. We use a bank of resonating feed backward comb filters with different time lags (τ), defined as:

$$y(t, \tau) = x(t) + \alpha * y(t - \tau), \quad (1)$$

Each comb filter adds a scaled (by factor α) and delayed (with lag τ) version of its own output $y(t)$ to the input signal $x(t)$ with t denoting the time frame index.

¹ Because of this smoothing the beat activations do not reflect probabilities any more (and they may exceed the value of 1), but this does not harm the overall interpretation and usefulness.

3.3.1 Lag Range Definition

For the individual bands of the comb filter bank we use a linear spacing of the lags with the minimum and maximum delays calculated as:

$$\begin{aligned}\tau_{min} &= \lceil 60 * fps / bpm_{max} \rceil \\ \tau_{max} &= \lceil 60 * fps / bpm_{min} \rceil\end{aligned}\quad (2)$$

with fps representing the frame rate of the system given in frames per second and the minimum and maximum tempi bpm_{min} and bpm_{max} given in beats per minute. We found the tempo range of $[40, 250]$ bpm to perform best on the validation set.

3.3.2 Scaling Factor Definition

Scheirer [26] found it beneficial to use different scaling factors $\alpha(\tau)$ for the individual comb filter bands. He defines them such that the individual filters have the same half-energy time. Klapuri [17] also uses filters with exponentially decaying pulse response, but sets the scaling factor such that the response decays to half after a defined time of 3 seconds.

Contrary to these findings, we use a single value for all filter lags, which is set to $\alpha = 0.79$. The reason that a single value works better for this system may lay in the fact that we sum all peaks of the filters. With a fixed scaling factor, the resonance of filters with smaller lags tend to decay faster, but they also produce more peaks, hence leading to a more “balanced” histogram.

3.3.3 Histogram Building

After smoothing the neural network output and processing it with the comb filter, we build a weighted histogram $H(\tau)$ from the output $y(t, \tau)$ by simply summing the activations of the individual comb filters (over all frames) if this filter produced the highest peak at the given time frame:

$$\begin{aligned}H(\tau) &= \sum_{t=0}^T y(t, \tau) * I(\tau, \arg \max_{\tau} y(t, \tau)) \\ I(a, b) &= \begin{cases} 1 & \text{if } a \equiv b \\ 0 & \text{otherwise} \end{cases}\end{aligned}\quad (3)$$

with t denoting the time frame index, T the total number of frames, and τ the filter delays.

The bins of the weighted histogram correspond to the time lags τ and the bin heights represent the number of frames where the corresponding filter has a maximum at this delay, weighted by the activations of the comb filter. This weighting has the advantage that it favours filters which resonate at lags which correspond to intervals with highly probable beat positions (i.e. high values of the beat activation function) over those which are less probable. Figure 2d illustrates the output of the comb filter bank, Figure 2e the weighted maxima which are used to build the weighted histogram shown as the dotted line in Figure 2f.

3.3.4 Histogram Smoothing

Music almost always contains tempo fluctuations – at least with regard to the frame rate of the system. Even stable tempi result in weights being split between two or more histogram bins. Therefore we combine bins before reporting the final tempo.

Our approach simply smooths the histogram by convolving it with a Hamming window with a width of seven bins, similar to [25]. Depending on the bin index (corresponding to the filter lag τ), a fixed width results in different tempo deviations, ranging from -7% to $+8\%$ for a lag of $\tau = 24$ (corresponding to 250 bpm) to -2% to $+2.9\%$ for a lag of $\tau = 40$ (i.e. 40 bpm). Although this allows a greater deviation for higher tempi, we found no improvement over choosing the size of the smoothing window as a function of the tempo. Figure 2f shows the smoothed histogram as the solid line.

3.3.5 Peak Selection

The histogram shows peaks at the different tempi of the musical piece. Again, previous works put much effort into this stage to select the peak with the strongest perceptual strength, ranging from simple rules driven by heuristics [25] over GMM regression based solutions [24] to utilising a support vector machine (SVM) [10, 25] or decision trees [25]. In order to keep our approach as simple as possible, we simply select the highest peak of the smoothed histogram as our final tempo.

4. EVALUATION

To assess the performance of the proposed system we compare it to an autocorrelation based tempo estimation method as described in [1], which operates on the same beat activation function obtained with the neural network described in Section 3.2. The algorithms of Gkiokas [9], Percival [25], Klapuri [17], Oliveira [23], and Davies [4] were chosen as additional reference systems based on their availability and overall performance.

For a short description of these algorithms, please refer to Section 2.

All of the algorithms were used in their default configuration, except the system of Oliveira [23], which we operated in offline mode with an induction length of 100 seconds, because it yielded significantly better results.² It should be noted however, that this mode results in a reduced tempo search range of 81-160 bpm, which can lead to biased results in favour of datasets in this tempo range.

Following [29] and [25] we perform statistical tests of our results compared to the others with McNemar’s test using a significance value of $p < 0.01$.

4.1 Evaluation Metrics

Since humans perceive tempo and rhythm subjectively, there is no single best tempo estimate. For example, the perceived tempo can be a multiple or fraction of the tempo given by the score of the piece. This is also known as

²This corresponds to: `ibtf -off -i auto-regen -t 100`

the tempo octave problem. Therefore, two evaluation measures are used in the literature: *Accuracy 1* considers only the single annotated tempo for the evaluation, whereas *Accuracy 2* also includes integer multiples or fractions of the annotated tempo. Since the data that we use also contains music in ternary meter, we do not only add double and half tempo annotations, but also triple and third tempo. In line with most other publications we report accuracy values which denote the algorithms' ability to correctly estimate the tempo of the musical piece with less than 4% deviation from the annotated ground truth.

4.2 Datasets

We use a total of ten datasets to evaluate the performance of our algorithm. Table 1 lists some statistics of the datasets. Datasets marked with an asterisk (*) were used to train the neural networks with 8-fold cross validation as described in Section 3.2.1.

For all sets with beat annotations available (Ballroom, Hainsworth, SMC, Beatles, RWC, HJDB), we generated the tempo annotations as the median of the inter beat intervals. For the HJDB set (which is in 4/4 meter), we first derived the beat positions from the downbeat annotations before inferring the tempo ground truth. For all other sets we use the provided tempo annotations and – where applicable – the corrected annotations from [25].

Dataset	# files	length	annotations
Ballroom [12, 19] *	685 ³	5h 57m	beats
Hainsworth [13] *	222	3h 19m	beats
SMC [16] *	217	2h 25m	beats
Klapuri [17]	474	7h 22m	beats
GTZAN [25, 27]	999	8h 20m	tempo
Songs [12]	465	2h 35m	tempo
Beatles [5]	180	8h 9m	beats
ACM Mirum [21, 24]	1410	15h 5m	tempo
RWC Popular [11]	100	6h 47m	beats
HJDB [15]	235	3h 19m	downbeats
total	4987	63h 17m	

Table 1: Overview of the datasets used for evaluation.

4.3 Results & Discussion

Table 2 lists the results of the proposed algorithm compared to the reference systems. The results (of our algorithm) reported on the Ballroom, Hainsworth and SMC set are obtained with 8-fold cross-validation, since these datasets were used to train the neural network. Although this is a technically correct evaluation, it can lead to biased results, since the system knows, e.g. about ballroom music and its features in general and thus has an advantage over the other systems. It is thus no surprise that the proposed system outperforms the others on these sets.

³We removed the 13 duplicates identified by Bob Sturm: http://media.aau.dk/null_space_pursuits/2014/01/ballroom-dataset.html

Nonetheless, the new system outperforms the autocorrelation based tempo estimation method operating on the very same neural network output in almost all cases. This clearly shows the advantage of the resonating comb filters, which are less prone to single missing or misaligned peaks in the beat activation function, due to their recurrent nature and the fact that they also resonate on fractions and multiples of the dominant tempo.

The results for the other datasets reflect the algorithm's ability to estimate the tempo of a completely unknown signal without tuning any of the parameters. It can be seen that no single system performs best on all datasets. Our proposed system performs state-of-the-art (i.e. no other algorithm is statistically significantly better) in all but the HJDB set w.r.t. *Accuracy 2*. We even outperform most of the other methods in *Accuracy 1*, which highlights the algorithm's ability to not only capture a meaningful tempo, but also choose the correct tempo octave.

An inspection of incorrectly detected tempi in the HJDB set showed that the algorithm's histogram usually has a peak at the correct tempo but that this peak is not the highest. The reason lays in the fact that this set contains music with breakbeats and strong syncopation. Unfortunately, the neural network often identifies these syncopated notes as beats. Contrary to single or infrequently misaligned beats, the comb filter is not able to correct regularly recurring misalignments. E.g. in drum & bass music, where the bass drum usually falls on the offbeat between the third and fourth beat, this leads to additional peaks in the histogram corresponding to 0.5 and 1.5 times the beat interval, and a much lower peak at the correct position. Since we do not perform intelligent clustering of the histogram peaks, often the rate of the downbeats is reported, which results in a tempo which is not covered by the *Accuracy 2* measure any more.

4.4 MIREX Evaluation

We submitted the algorithm to last year's MIREX evaluation. ⁴ Performance is tested on a hidden set of 140 files with a total length of 1 hour and 10 minutes. The tempo evaluation used for MIREX is different, because for each song the two most dominant tempi are annotated. MIREX uses the following three evaluation metrics: *P-Score* [22] and the percentage of files for which *at least one* or *both* of the annotated tempi was identified correctly within a maximum allowed deviation of $\pm 8\%$ from the ground truth annotations. Since MIREX requires the algorithms to report two tempi with a relative strength, we adopted the peak-picking strategy outlined in Section 3.3.5 to simply report the two highest peaks.

Table 3 gives an overview of the five best performing algorithms (of different authors) over all years the MIREX tempo estimation task is run, together with results for algorithms also used for evaluation in the previous section.

Our algorithm ranked first in last year's MIREX evaluation and achieved the highest *P-Score* and *at least one tempo reported correctly* performance ever. The best per-

⁴<http://nema.lis.illinois.edu/nema-out/mirex2014/results/ate/>

	NEW	Böck [1]	Gkiokas [9]	Percival [25]	Klapuri [17]	IBT [23]	Davies [4]
<i>Accuracy 1</i>							
Ballroom [12, 19]	0.950 †	0.639†	0.625-	0.653-	0.642-	0.651-	0.709-
Hainsworth [13]	0.847 †	0.541†	0.667-	0.721-	0.752-	0.698-	0.739-
SMC [16]	0.512 †	0.442†	0.346-	0.267-	0.189-	0.166-	0.152-
Klapuri [17]	0.789	0.502-	0.741	0.732	0.768	0.724-	0.692-
GTZAN [25]	0.668	0.601-	0.716-	0.754 +	0.704+	0.599-	0.582-
Songs [12]	0.477	0.570+	0.570+	0.611 +	0.585+	0.486	0.424
Beatles [5]	0.850	0.700-	0.778	0.811	0.789	0.767	0.761-
ACM Mirum [21, 24]	0.741	0.540-	0.725	0.733	0.679-	0.621-	0.646-
RWC Popular [11]	0.600	0.450	0.900 +	0.810+	0.770	0.750	0.770+
HJDB [14]	0.796	0.434-	0.783	0.285-	0.494-	0.911 +	0.706
Dataset average	0.721	0.543	0.563	0.638	0.636	0.637	0.617
Total average	0.734	0.560-	0.685-	0.677-	0.658-	0.623-	0.618-
<i>Accuracy 2</i>							
Ballroom [12, 19]	1.000 †	0.997†	0.981	0.953-	0.921-	0.921-	0.974
Hainsworth [13]	0.941 †	0.910†	0.887	0.901	0.869	0.802-	0.878
SMC [16]	0.673 †	0.599†	0.512-	0.438-	0.438-	0.359-	0.415-
Klapuri [17]	0.937	0.907-	0.954	0.937	0.918	0.880-	0.924
GTZAN [25]	0.950	0.942	0.938	0.925-	0.923-	0.841-	0.922-
Songs [12]	0.933	0.918	0.910	0.865-	0.910	0.791-	0.875-
Beatles [5]	0.983	0.967	0.978	0.989	0.928	0.883	0.978
ACM Mirum [21, 24]	0.976	0.958-	0.979	0.972	0.967	0.915-	0.975
RWC Popular [11]	0.950	0.940	1.000	1.000	0.990	0.980	1.000
HJDB [14]	0.868	0.851	0.911	1.000 +	0.864	0.991+	1.000 +
Dataset average	0.919	0.899	0.916	0.896	0.871	0.837	0.893
Total average	0.946	0.929-	0.935-	0.923-	0.909-	0.861-	0.923-

Table 2: *Accuracy 1* and *Accuracy 2* results for different datasets and algorithms, with best results marked in bold and + and - denoting statistical significance compared to our results. † denote values obtained with 8-fold cross validation.

<i>Algorithm</i>	P-Score	≥ 1 tempo	both tempi
NEW	0.876	0.993	0.629
Elowsson [7]	0.857	0.943	0.693
Gkiokas [9]	0.829	0.943	0.621
Wu [28]	0.826	0.957	0.550
Lartillot [20]	0.816	0.921	0.571
Klapuri [17]	0.806	0.943	0.614
Böck [1]	0.798	0.957	0.564
Davies [4]	0.776	0.929	0.457

Table 3: Results on the McKinney test collection used for the MIREX evaluation.

forming algorithm for the *both tempi correct* evaluation was the one submitted by Elowsson [7] in 2013, which explicitly models the speed of the music and thus has a much higher chance to report the two annotated tempi which are inferred from human beat tapping.

5. CONCLUSION

The presented tempo estimation algorithm based on recurrent neural networks and resonating comb filters is able to perform state-of-the-art or outperforms existing algorithms on all but one datasets investigated. Based on the high *Ac-*

curacy 2 score, which also considers integer multiples and fractions of the annotated ground truth tempo, it can be concluded that the system is able to capture a meaningful tempo in almost all cases.

Additionally, we outperform many existing algorithms w.r.t. *Accuracy 1* which suggests that it is advantageous to use a musically more meaningful representation than just the onset strength of the signal – even if split into multiple accent bands – as an input for a bank of resonating comb filters.

In future, we want to investigate methods of perceptually clustering the peaks of the histogram to report the most relevant tempo, as this has been identified to be the main problem of the new algorithm when dealing with very synopated music. We believe that this should increase the *Accuracy 1* performance considerably.

The source code and additional resources can be found at: <http://www.cp.jku.at/people/boeck/ISMIR2015.html>.

6. ACKNOWLEDGMENTS

This work is supported by the European Union Seventh Framework Programme FP7 / 2007-2013 through the GiantSteps project (grant agreement no. 610591) and the Austrian Science Fund (FWF) project Z159. We would like to thank the authors of the other algorithms for sharing their code or making it publicly available.

7. REFERENCES

- [1] S. Böck and M. Schedl. Enhanced Beat Tracking with Context-Aware Neural Networks. In *Proc. of the 14th International Conference on Digital Audio Effects (DAFx)*, pages 135–139, Paris, France, 2011.
- [2] S. Böck, F. Krebs, and G. Widmer. A multi-model approach to beat tracking considering heterogeneous music styles. In *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, Taipei, Taiwan, 2014.
- [3] A. T. Cemgil, B. Kappen, P. Desain, and H. Honing. On tempo tracking: Tempogram Representation and Kalman filtering. *Journal of New Music Research*, 28:4:259–273, 2001.
- [4] M. E. P. Davies and M. D. Plumbley. Context-dependent beat tracking of musical audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(3):1009–1020, 2007.
- [5] M. E. P. Davies, N. Degara, and M. D. Plumbley. Evaluation methods for musical audio beat tracking algorithms. Technical Report C4DM-TR-09-06, Centre for Digital Music, Queen Mary University of London, 2009.
- [6] S. Dixon. Automatic extraction of tempo and beat from expressive performances. *Journal of New Music Research*, 30:39–58, 2001.
- [7] A. Elowsson, A. Friberg, G. Madison, and J. Paulin. Modelling the speed of music using features from harmonic/percussive separated audio. In *Proc. of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, Curitiba, Brazil, 2013.
- [8] M. Gainza and E. Coyle. Tempo detection using a hybrid multiband approach. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(1):57–68, 2011.
- [9] A. Gkiokas, V. Katsourous, G. Carayannis, and T. Stafylakis. Music tempo estimation and beat tracking by applying source separation and metrical relations. In *Proc. of the 37th International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 421–424, Kyoto, Japan, 2012.
- [10] A. Gkiokas, V. Katsourous, and G. Carayannis. Reducing Tempo Octave Errors by Periodicity Vector Coding And SVM Learning. In *Proc. of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 301–306, Porto, Portugal, 2012.
- [11] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC Music Database: Popular, Classical, and Jazz Music Databases. In *Proceedings of the 3rd International Conference on Music Information Retrieval (ISMIR 2002)*, pages 287–288, Paris, France, 2002.
- [12] F. Gouyon, A. Klapuri, S. Dixon, M. Alonso, G. Tzanetakis, C. Uhle, and P. Cano. An experimental comparison of audio tempo induction algorithms. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5):1832–1844, 2006.
- [13] S. Hainsworth and M. Macleod. Particle filtering applied to musical tempo tracking. *EURASIP Journal on Applied Signal Processing*, 15:2385–2395, 2004.
- [14] J. Hockman and I. Fujinaga. Fast vs slow: Learning tempo octaves from user data. In *Proc. of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 231–236, Utrecht, Netherlands, 2010.
- [15] J. Hockman, M. E. Davies, and I. Fujinaga. One in the jungle: Downbeat detection in hardcore, jungle, and drum and bass. In *Proc. of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 169–174, Porto, Portugal, 2012.
- [16] A. Holzapfel, M. E. P. Davies, J. R. Zapata, J. L. Oliveira, and F. Gouyon. Selective sampling for beat tracking evaluation. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(9):2539–2548, 2012.
- [17] A. P. Klapuri, A. J. Eronen, and J. T. Astola. Analysis of the meter of acoustic musical signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):342–355, 2006.
- [18] F. Korzeniowski, S. Böck, and G. Widmer. Probabilistic extraction of beat positions from a beat activation function. In *Proc. of the 15th International Society for Music Information Retrieval Conference (ISMIR)*, pages 513–518, Taipei, Taiwan, 2014.
- [19] F. Krebs, S. Böck, and G. Widmer. Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *Proc. of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, pages 227–232, Curitiba, Brazil, 2013.
- [20] O. Lartillot, D. Cereghetti, K. Eliard, W. J. Trost, M.-A. Rappapaz, and D. Grandjean. Estimating tempo and metrical features by tracking the whole metrical hierarchy. In *Proc. of the 3rd International Conference on Music & Emotion (ICME)*, Jyväskylä, Finland, 2013.
- [21] M. Levy. Improving perceptual tempo estimation with crowd-sourced annotations. In *Proc. of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 317–322, Miami, USA, 2011.
- [22] M. F. McKinney, D. Moelants, M. E. P. Davies, and A. Klapuri. Evaluation of Audio Beat Tracking and Music Tempo Extraction Algorithms. *Journal of New Music Research*, 36(1):1–16, 2007.
- [23] J. Oliveira, F. Gouyon, L. G. Martins, and L. P. Reis. IBT: a real-time tempo and beat tracking system. In *Proc. of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, Utrecht, Netherlands, 2010.
- [24] G. Peeters and J. Flocon-Cholet. Perceptual tempo estimation using GMM-regression. In *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, pages 45–50, 2012.
- [25] G. Percival and G. Tzanetakis. Streamlined tempo estimation based on autocorrelation and cross-correlation with pulses. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(12):1765–1776, 2014.
- [26] E. D. Scheirer. Tempo and beat analysis of acoustic musical signals. *The Journal of the Acoustical Society of America*, 103(1):588–601, 1998.
- [27] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, 2002.
- [28] F.-H. F. Wu and J.-S. R. Jang. A supervised learning method for tempo estimation of musical audio. In *22nd Mediterranean Conference of Control and Automation (MED)*, pages 599–604, Palermo, Italy, 2014.
- [29] J. Zapata and E. Gómez. Comparative evaluation and combination of audio tempo estimation approaches. In A. E. Society, editor, *AES 42nd Conference on Semantic Audio*, Ilmenau, Germany, 2011. Audio Engineering Society, Audio Engineering Society.

MUSICOLOGY OF EARLY MUSIC WITH EUROPEANA TOOLS AND SERVICES

Erik Duval¹, Marnix van Berchum², Anja Jentsch³,
Gonzalo Alberto Parra Chico¹, Andreas Drakos⁴

¹erik.duval@cs.kuleuven.be, Dept. of Computer Science, KU Leuven, B

²marnix.van.berchum@dans.knaw.nl KNAW-DANS, Utrecht University, NL

³anja.jentsch@okfn.org Open Knowledge Foundation, D

⁴ AgroKnow, GR

ABSTRACT

The Europeana repository hosts large collections of digitized music manuscripts and prints. This paper investigates how tools and services for this repository can enable Early Music musicologists to carry out their research in a more effective or efficient way, or to carry out research that is impossible to do without such tools or services. We report on the methodology, user-centered development of a suite of tools that we have integrated loosely, in order to experiment with this specific target audience and an evaluation of the impact that such tools may have on how these musicologists carry out their research. Positive feedback relates to the automation of data sharing between the loosely coupled tools and support for an integrated workflow. Participants in this study wanted to have the ability to work not only with individual items, but also with collections of such items. The use of search facets to filter, and visualization around time and place were positively evaluated, as was the use of Optical Music Recognition and computer-supported analysis of music scores. The musicologists were not convinced of the value of activity streams. They also wanted a less strictly linear organization of their workflow and the ability to not only consume items from the repository, but to also push their research results back into the Europeana repository.

1. INTRODUCTION AND BACKGROUND

The basic aim of the work presented in this paper is to develop services and tools that leverage content in the Europeana Cloud for researchers in digital humanities [4]. In a first year of experimentation, we focused on content in the Wittgenstein archives at the University of Bergen and the Axiom philosophy group at the VU University Amsterdam [5]. In this paper, we report on experimentation in a second year of the project, where we targeted a research community of musicologists that focus on Early Music.

It is important to note that the Europeana Cloud project has a much wider scope: it is concerned with migrating the backend technology of Europeana to a cloud-

based infrastructure. The focus of our work is to demonstrate that this technical development enables new tools and services that make it possible for researchers in digital humanities (in the specific case of the work presented in this paper: researchers in Early Music) to either carry out their existing research in a more effective or efficient way, or to carry out research work that is impossible without such tools and services, at least in practical terms, for instance because it would involve too much manual tedious human labor.

In the early phase of the project, as the cloud-based services are still under development, we investigate this issue of added value by loosely integrating existing tools and services accessing the original Europeana services and other suitable services, and by imitating the workflow of the Europeana research platform, which is still under development.

2. RESEARCH GOAL AND METHODOLOGY

2.1 Research questions

In this paper, we address the following research questions:

1. What are the main problems for digital musicologists whose research focuses on Early Music?
2. How can we address these problems and demonstrate the potential added value of cloud-based tools and services on top of large repositories of content like Europeana for Early Music musicologists?

2.2 Methodology

Our basic methodology is User Centered Design [1]. The users of this iteration were musicologists working on Early Music (up to and including Monteverdi). A small group (5 persons) was selected from within the network of the authors. Besides their focus on Early Music, the musicologists in the group share an affinity with technology, and to a different degree are all involved in applying technology to their research practice.

As designers and developers, we had regular formative evaluation sessions over Skype or Google Hangout with the musicologists. (In fact, this worked surprisingly well and allowed for many more regular meetings than we could have organized in more traditional settings with such a diverse, busy and geographically distributed group of participants.) We also had a face-to-face meeting at the



© Erik Duval, Marnix van Berchum, Anja Jentsch,
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Erik Duval, Marnix van Berchum, Anja Jentsch, Gonzalo Alberto Parra Chico and Andreas Drakos. "Musicology of Early Music with Europeana tools and services", 16th International Society for Music Information Retrieval Conference, 2015.

end of the yearly development cycle, for a more in-depth evaluation (see section 6).

In initial meetings the musicologists discussed with us the workflow, computational tools, and content that they currently use.

It is important to note that the evaluation sessions focused on usefulness and usability-in-the-large, i.e. on whether or not the foreseen tools and associated research methodology would actually be of any substantial added value to the researchers involved. We wanted, more specifically, to find out whether our approach could help them to actually change the way they work, whether such an approach would address problems that they may or may not be aware of in their current way of working, etc. Only to a much lesser extent were we interested in finding out whether the Early Music researchers can carry out their current way of working in a more efficient way with our tools and methodology.

3. RELATED WORK

In the past decades, the musicology community in general has been actively involved in the use and development of digital tools for enhancing musicological research. The scholarly study of Early Music is no exception, focusing on very specific problems from this period of music history, while still making use of generic solutions. The development of encoded music formats has been very important, opening up opportunities for musicologists to make use of and analyze machine-readable scores [18]. Seminal work on music encoding is carried out from the eighties onwards, culminating for now in more recent work on how full digital, critical editions of Early Music could be conceived. Further proof of the affinity of the Early Music community can be found in a special issue of the journal *Early Music* (i.e. Volume xlii (2014), No. 4). Whereas some research has focused on Optical Music Recognition (OMR) for automated metadata generation [11], we rely on metadata from repositories of musical sources (manuscripts, prints) in Europeana and apply OMR techniques in a later step in order to generate a machine readable music encoding for analysis (see section 5.5). In that sense, the scope and goal of the work presented here is more similar to [6], though we focus specifically on Early Music and a User-Centered Design approach for end user tool design and development (section 2.2). An outcome of this approach is that we provide geospatial and time based visualization of search results, rather than a more conventional list of search results, as used in for instance [11]. In fact, we believe that visual approaches to music access remain underexplored, despite some work like [16] and [19]. Our work is a bit different from this earlier work on visualization in that it focuses more on visualizations based on geospatial and time based characteristics of music rather than on visualizing clusters of related music.

The User-Centered Design approach, which is also central to the work presented in this paper, found its way already in the emerging field of ‘digital musicology’ [2][3]

but our focus is on leveraging the content from large-scale repositories for musicology.

4. MAIN PROBLEMS FOR MUSICOLOGISTS

At the initial stage of our work, we identified the following four core problems for the musicologists in our discussions with them:

1. Difficulty of creating the data and metadata needed: the creation of encoded music scores of Early Music (i.e. ‘musical data’) is a laborious task, which is often carried out with proprietary software packages not suited for the particular types of music notation from this period. Likewise, the metadata on these scores, their original sources, the composers etc. are locked into paper publications and not easily transformed into digital format.
2. Lack of digital corpora with music scores: there are some repositories with music scores for Early Music, like for example CMME (<http://www.cmme.org>), ECOLM (<http://www.ecolm.org>), the Josquin Research Project (<http://josquin.stanford.edu>) and SIMSSA (<http://www.simssa.ca>) [6], but they are fragmented and it is tedious and time-consuming to go through the different repositories (each with their own query facilities) and do a systematic search for a particular composer or theme.
3. Information exchange and linking of data when working with different tools: although there are specific tools to process music scores, they do not inter-operate and it is again quite tedious and time-consuming to apply different tools on the same content and then to integrate the results of the different tools.
4. Retrieval and analysis of contextual information about the music scores, from bibliographical and historical databases, like the Oxford Music Online (<http://www.oxfordmusiconline.com/>) or RILM (<http://www.rilm.org>).

As will become clear in the remainder of this paper, we eventually succeeded in addressing the 1st, 2nd and 3rd problem listed above.

5. TOOL SETUP

5.1 Introduction

In order to investigate how technology can help the musicologists with these problems, we designed, created, integrated and evaluated a set of prototype tools that extends the toolset we prepared for the philosophers the year before. The complete toolset consists of (see Figure 1):

- Ariadne Finder (section 5.2): this tool, personalized for musicologists, helps researchers search and find content coming from Europeana and other sources in a simple and integrated way - the intent is that this tool addresses problem 2 mentioned above;

- TimeMapper (section 5.3): this integrated tool visualizes the search results from the Ariadne finder on a timeline and an interactive map, in order to enable the musicologists to further filter the content and get a better overview of the different resources found on Europeana (<http://timemapper.okfnlabs.org>);
- Activity Stream (section 5.4): this service, integrated in all the tools, captures and presents the different actions carried out by the users in their interactions with the tools;
- Aruspix (section 5.5): this is an optical music recognition (OMR) tool which transforms prints of Early Music scores into MEI [13];
- Music21 (section 5.6): this is a Python-based set of tools for analysing music encoded as XML (<http://web.mit.edu/music21/>) [6].

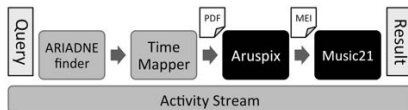


Figure 1: Schema of interconnected tools;

5.2 Ariadne Finder

A series of meetings with the musicology researchers enabled us to identify the content collections of interest. To the Europeana base collection, we added the resources from RISM (<http://www.rism.info/>), and integrated them in the Finder. RISM is a well-known and extensively used inventory of musical sources. The abbreviations of library sigla used in RISM, have an authoritative character within musicology, and can be used as a controlled vocabulary in a digital environment.

After the first year experimentation, we simplified the user interface of the Finder by removing some predefined categories from the home screen. Instead, we made a list of four search facets (i.e. provider, media type, language, and year) available on the first screen with the search results.

The integration of the RISM collection was a great challenge: the data covered by RISM (metadata on primary musical sources) are heterogeneous and quite different from the ones provided by Europeana. To allow the integration with the Finder backend and to enable the visualisation of search results in a uniform way, transformation of the metadata to an internal format was required. Moreover, linking to the actual resource was not possible, since RISM provides metadata on the current (physical) holding of the sources, and does not provide links to the digitized versions of the sources.

The Finder is used as the ‘baseline’ tool for the integration of the other tools, listed below. Both the Activity Stream and the TimeMapper are integrated in the Finder to see the past user activities (i.e. searches) and to visualise search results respectively. When viewing an individ-

ual search result, the connection to Music21, through Aruspix, is also available.

In Figure 2, the listing of the search results is shown, with the facets on the left that can be used to further refine the search. Finally, Figure 3 shows how an individual search result is displayed to the user, with the links to the functionality of Aruspix and Music21.

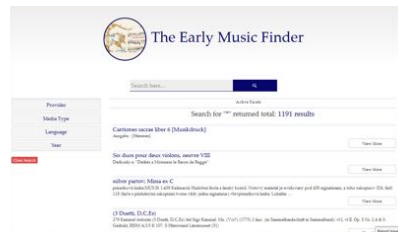


Figure 2 : Search results in the Finder



Figure 3 : Individual search result in the Finder

The Ariadne Finder for the Musicologists group can be accessed at <http://greenlearningnetwork.com/cmme-finder/>.

5.3 TimeMapper

Europeana provides a variety of metadata for its resources, including thumbnail images, geo-coordinates and time information. TimeMapper visualizes the temporal and geographical characteristics of resources.

TimeMapper is a data visualization tool that allows for the creation of timelines and timemaps using Google spreadsheets (<http://timemapper.okfnlabs.org>). While the Finder provides the user with a faceted search for Europeana resources, it might still be difficult to navigate through large amounts of search results. We integrated TimeMapper in our tool chain to provide an interactive geo-spatial visualization of the search results. This enables users to quickly navigate the metadata and to order resources on the basis of time and place of publication. In

this way, they can more easily identify resources worth studying in more detail.

Figure 4 shows the TimeMapper when drilling down into resources that match the keyword “Gardano”. TimeMapper is available under the MIT licence. The tool can be accessed via the Ariadne Finder button labelled “View in TimeMapper”.

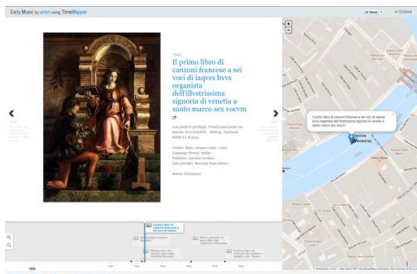


Figure 4: TimeMapper showing resources published by Gardano

5.4 Activity Stream

Based on our earlier work on community reading awareness **Error! Reference source not found.** and supporting the Science 2.0 idea of enhancing collaboration among researchers [17], we have designed, developed and deployed a web application called the “Activity Stream (AS)”, enabling researchers to share their work related activities within a community. More specifically in the context of the Early Music musicologists, the application aggregates “search” and “visualize” activities, and makes researchers aware of what their peers are currently working on.

In the first prototype, the AS presented information about “searches” that were carried out with the Ariadne Finder and terms that were “visualized” using the TimeMapper, as illustrated by Figure 5. The activities in the stream are structured as: Actor | verb | (Object). For example, “User from GR” | “has searched” | “Bolzano”. For the musicologists, two new activities were added to the activity stream: interpretation and processing. These represent the usage of the Aruspix and Music21 components (see below).



Figure 5: Main screen of the activity stream

The Activity Stream is implemented as a web application (using HTML and JavaScript) and deployed using the Google App Engine (GAE). Together with the terms used to perform a search or visualization, a link to the tool showing the outcome of that action is provided. Also, in order to provide users the flexibility to filter activities, tool grouping was added to the application. For instance, by clicking on the tool’s name (e.g.: Finder or TimeMapper) the user can consult the stream of activities from that tool only.

The Activity Stream allows us to digest different events sent from different tools (via REST services) used by researchers, but also provides the possibility to embed these in other software components. For example, the application supports RSS syndication as a passive notification system. Figure 6 illustrates the current activity sources and outlets.

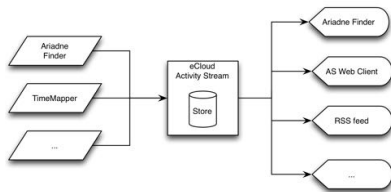


Figure 6: Information sources and destinations of the Activity Stream

5.5 Aruspix

Aruspix is an optical music recognition (OMR) tool that scans early music prints, transcribes them and encodes them into the MEI standard [9][10][15].

While there are other OMR tools available, mainly for music in common music notation, Aruspix is the only tool to our knowledge that can handle scores printed in the 16th and 17th centuries with movable typefaces. Such scores are often difficult to examine with existing superimposition and optical recognition software, as they present a number of specific layout and format problems and are quite often in a deteriorated state because of their age [12][13][14].

The printing techniques of that time mean that differences can exist between copies produced in the same print run, and comparison of these copies by superimposition can enable more accurate critical editions to be prepared. Digitizing the scores through optical recognition can enable us to collate different editions regardless of layout, and is also useful in for instance the preparation of digital music libraries.

For Europeana Cloud, we use the command line version of Aruspix that automatically converts digital scans of scores to MEI files in a page-wise fashion. We then need to combine the pages into a single score again.

Moreover, the MEI version being used by Aruspix is a new and not yet standardized one[13].

Since Music21 (see next section) needs MEI files that use the 2012 or 2013 specification, we developed an XSLT program to transform the MEI files that Aruspix delivers into this newer format.

The command line version sends requested score transcriptions to the Music21 service for further analysis (Section 5.6). Furthermore, it sends activity on transcribed scores to the Activity Stream (Section 5.4).

5.6 Music21

Music21 is a Python-based object-oriented toolkit for computer-aided musicology that allows music information, extraction and generation, together with music notation editing and scripting in symbolic (score-based) forms (<http://web.mit.edu/music21/>)[6]. The toolkit is able to import different formats, such as MusicXML and MEI.

We extended the Music21 web application module in order to provide parsing and processing requests to a Music21 installation running on a server. In the workflow, Music21 is used after the Aruspix service has created an MEI version of a score. With an MEI file, a specific set of actions becomes available to the musicologists in order to support them with the analysis of the music involved: calculation of 'Parts and Measures', calculation of the 'Pitch ranges' and requesting the 'legal melodic intervals' of a score.

6. EVALUATION

6.1 General evaluation

To start the discussion, the complete workflow of tools was presented to the musicologists. Afterwards, questions were asked regarding the usefulness of their current tool setup. In general, the participants agreed that the way in which the tools support the research process is helpful. The connection of existing tools (optical music recognition and processing of encoded scores) and automating the process of data sharing between these tools is of great value for them, as it saves them time with their research tasks, compared with using the tools individually. Actually, some of the musicologists had not been able to manually feed the output of one tool as input to the next tool in the workflow.

While the participants found the overall workflow useful, they were also interested in details about specific parts of it. Some of them suggested that, in some cases, just one or two tools are more relevant for their research (e.g. converting a score into a computer readable format or importing their own encoded scores for processing with Music21). This is mainly related to their very varied technical background and research goals. Some of the participants are computational musicologists that regularly use tools like Music21, while others are more traditional musicologists that work with the original sources and have very limited digital research experience.

The participants agreed with the added value of the loosely integrated workflow while doing research on a single item (score), but also observed that the workflow could be automated for use at a larger scale (e.g. a large dataset of scores of a specific period or region). Such automation could be of great value in order to answer research questions about a complete collection or in order to generate new questions for such a collection.

6.2 Ariadne Finder, TimeMapper and Activity Stream

After the musicologists discussed the overall workflow, the loose coupling and setup of tools, they were prompted to assess the tools on an individual level.

From the set of tools adapted from the experimentation the year before with the philosophers [6], the TimeMapper was considered the most interesting and relevant for musicology research. In its current form, the tool provides a visualization of scores based on location and year of print. The participants suggested extending the functionality of the tool, for example with the use of more information than just the data of publication of the prints (e.g. include the information gathered in the Music21 tools, like parallel fifths, valid melodies, or other species counterpoints of a score or measure) or the possibility to compare different timelines that represent results for different search terms. This feedback basically confirms the relevance and usefulness of information visualization techniques in general for musicology research [16][19].

The Finder was mostly seen as a tool that provides existing functionality, similar to what other search engines provide, though the musicologists acknowledged the value of having facets to filter the result set. They suggested to personalize facets to terms that are closer to musicologist research practice, for example, to use 'printed books', 'manuscripts', 'single pieces' instead of 'image' or 'text' classification.

The musicologists were more critical about the usefulness of the Activity Stream (AS) in their research activities. They were not sure that the current actions are relevant for them or even which alternative kinds of activities might be useful to be displayed in the tool. They mostly perceived the AS as an interesting communication device or as a source of information that is comparable to what is common in a Social Network (like Facebook, or more specific for research, like <https://www.academia.edu> or <https://www.researchgate.net/>). The participants suggested functionality to enhance the perceived usefulness of the stream, such as a search for specific activities, the possibility to aggregate activities in order to obtain statistics from them, and the possibility to store results for later use.

Participants also suggested other interesting ways to connect the tools, instead of only having a linear approach, as in the current setup. For example, they mentioned that it would be interesting to be able to take the output of Music21 (e.g. parallel fifths of a score) and map

the results, based on their location, with the TimeMapper. This can provide an overview of specific score characteristics and relate them to a particular location.

6.3 Aruspix and Music21

While the Music21 version included in our workflow does not have a visual frontend for the users, the musicologists acknowledge its importance in the workflow. As mentioned, optical music recognition (OMR) is a crucial step for them [11][12][13][14]. Regarding the current output of this tool, the musicologists would appreciate to see the encoding result and the percentage of errors after the OMR process. While in other sciences, researchers are used to work with and accept a certain percentage of errors, these may not be well accepted in the musicology domain where there is much less of a tradition to work with data that include errors. Nevertheless, the musicologists appreciate what is happening behind the scenes and how good the obtained encoding is, and believe that the results could build trust from the user in the system. Moreover, information about errors can be used as a feedback mechanism for Aruspix: study participants mentioned that they wanted such a facility to be as simple as possible but at the same time complete enough to get the desired information.

The Music21 web interface was one of the most interesting tools for the musicologists. Besides the textual rendition of the analytical results, the participants would also like access to plots or statistics (e.g. note distribution), as these could be more helpful in order to identify characteristics of a score. Currently, the Music21 interface only supports a specific set of generic calculations and processes [6]. The participants would like to have the freedom to build their own analysis, via text or through a graphical user interface.

6.4 Other comments

During the face-to-face evaluation session, the participants provided suggestions about the tools and the workflow, but also about the underlying concepts. For example, some users suggested being able to push the generated encoded scores by Aruspix (MEI or MusicXML) back into the Europeana repository, so that we would use OMR technology to generate metadata, as in [12][14]. Likewise, results created with the Music21 toolkit could be considered as metadata of a particular composition, and as such could also be fed back into the Europeana Cloud repository. Such an approach would enable sharing intermediate research results with peers and a more Science2.0 approach to research [17].

While it was not the direct scope of our work, the participants made a number of suggestions for enhancing the specific usability of the tools and providing a nicer user interface overall.

Finally, the participants suggested additional tools or functionality to be considered. These included:

- Possibility to run batch processes, in order to get a broader overview of music characteristics of a set of scores.

- Support for playback mechanisms in Music21 (or Aruspix), in order to be able to validate and confirm the automatic encoding by listening to the result.
- Possibility to annotate directly into the digital version of a score.
- Possibility to create their own visualizations based on the data obtained from different tools, especially from the Music21 output.
- Inclusion of additional musicology resources, for example from <http://www.diamm.ac.uk/>.

7. CONCLUSION AND FUTURE WORK

Basically, the User-Centered Development process seems to work as intended: the target users positively evaluated the end result. An important issue for the next cycle is to connect the frontend tools for researchers with the actual backend infrastructure of Europeana Cloud, which has progressed into deployment while our work was taking place. This integration in the production system will enable us to work with more comprehensive content collections.

It is clear from the results that we obtained that there is substantial potential to support novel research methods on large-scale collections of music sources, using technologies like Optical Music Recognition, information visualization, loose coupling of tools, and flexible search. Our work illustrates how this can help researchers in Early Music to carry out existing research in more efficient and effective ways, and even address research questions that are hard or impossible to work on with more traditional means. As such, the potential for a Science2.0 approach to musicology is quite considerable.

8. ACKNOWLEDGEMENT

We gratefully acknowledge the support of the Europeana Cloud project, funded under the ICT Policy Support Programme (ICT PSP) as part of the Competitiveness and Innovation Framework Programme by the European Community (grant agreement no. 325091).

We are very grateful for the intensive discussions and detailed feedback from the following Early Music musicologists: Frans Wiering (Utrecht University), Reinier de Valk (City University London), Eliane Fankhauser (Utrecht University), Laurent Pugin (RISM Switzerland) and Peter van Kranenburg (Meertens Institute - KNAW)

9. REFERENCES

- [1] C. Abras, D. Maloney-Krichmar, and J. Preece, "User-centered design," *Encyclopedia of Human-Computer Interaction*, Vol. 37, No. 4, pp. 445–56, 2004.
- [2] A. Aljanaki, D. Bountouridis, J. A. Burgoyne, J. Van Balen, F. Wiering, H. Honing, and R. Veltkamp, "Designing games with a purpose for data collection in music research. Emotify and hooked: Two case studies," *Games and Learning Alliance*, pp. 29–40, 2014.
- [3] M. Barthet and S. Dixon, "Ethnographic observations of musicologists at the British Library: Implications for music information retrieval," *ISMIR11: Proc. of the 12th International Society for Music Information Retrieval Conference*, pp. 353–358, 2011.
- [4] A. Benardou, C. Dallas, and A. Dunning, "From Europeana Cloud to Europeana Research: The challenges of a community-driven platform exploiting Europeana content," *Digital Heritage. Progress in Cultural Heritage: Documentation, Preservation, and Protection*, pp. 802–810, 2014.
- [5] G. Parra, J. Klerkx and E. Duval, "What Should I Read Next?: Awareness of Relevant Publications through a Community of Practice," *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, pp. 2375–2376, 2013.
- [6] H. van den Berg, G. Parra, A. Jentzsch, A. Drakos and E. Duval, "Studying the History of Philosophical Ideas: Supporting Research Discovery, Navigation, and Awareness," *Proc. of the 14th Inter. Conf. on Knowledge Technologies and Data-driven Business*, 12:1-12:8, 2014.
- [7] M. S. Cuthbert and C. Ariza, "Music21: A toolkit for computer-aided musicology and symbolic music data," *ISMIR10: Proc. of the 11th International Conference on Music Information Retrieval*, pp. 637–642, 2010.
- [8] I. Fujinaga, A. Hankinson, and J. E. Cumming, "Introduction to SIMSSA (Single Interface for Music Score Searching and Analysis)," *DLfM14: Proc. of the 1st International Workshop on Digital Libraries for Musicology*, pp. 1–3, 2014.
- [9] A. T. Geertinger, and L. Pugin, "MEI for bridging the gap between music cataloguing and digital critical editions", *Die Tonkunst. Magazin für klassische Musik und Musikwissenschaft*, 5.3, pp. 289-294, 2011.
- [10] A. Hankinson, P. Roland, and I. Fujinaga, "The Music Encoding Initiative as a document-encoding framework". *ISMIR11: Proc. of the 12th International Society for Music Information Retrieval Conference*, pp. 293–298, 2011.
- [11] A. Hankinson, J. A. Burgoyne, G. Vigiensoni, A. Porter, J. Thompson, W. Liu, R. Chiu, and I. Fujinaga, "Digital document image retrieval using optical music recognition," *ISMIR12: Proc. of the 13th International Society for Music Information Retrieval Conference*, pp. 577–582, 2012.
- [12] L. Pugin, "Optical Music Recognition of Early Typographic Prints using Hidden Markov Models", *ISMIR06: Proc. Of the 7th International Conference on Music Information Retrieval*, pp. 53-56, 2006.
- [13] L. Pugin, J. Hockman, J.A. Burgoyne, and I. Fujinaga, "Gamera Versus Aruspix: Two Optical Music Recognition Approaches," *ISMIR08: Proc. of the 9th International Society for Music Information Retrieval Conference*, pp. 419-424, 2008.
- [14] L. Pugin and T. Crawford, "Evaluating OMR on the Early Music Online Collection," *ISMIR13: Proc. of the 14th International Society for Music Information Retrieval Conference*, pp. 439–44. 2013.
- [15] P. Roland, A. Hankinson, and L. Pugin, "Early music and the Music Encoding Initiative," *Early Music*, Vol. xlii, No. 4, pp. 605-611, 2014.
- [16] M. Schedl, P. Knees, K. Seyerlehner, and T. Pohle, "The CoMIRVA Toolkit for Visualizing Music-Related Data". *EUROVIS07: Proc of the 9th Eurographics/IEEE-VGTC Symp. on Visualization*, pp. 147-154, 2007.
- [17] B. Shneiderman, "Science 2.0," *Science*, Vol. 319, No. 5868, pp. 1349-1350, 2008.
- [18] J. Stinson and J. Stoessel. "Encoding medieval music notation for research," *Early Music*, Vol. xlii No. 4, pp. 613–617, 2014.
- [19] S. Stober and A. Nürnberger. "A multi-focus zoomable interface for multi-facet exploration of music collections," *7th International Symposium on Computer Music Modeling and Retrieval*, pp. 339–354, 2010.

SINGING VOICE SEPARATION FROM MONAURAL MUSIC BASED ON KERNEL BACK-FITTING USING BETA-ORDER SPECTRAL AMPLITUDE ESTIMATION

Hye-Seung Cho, Jun-Yong Lee, Hyoung-Gook Kim

Kwangwoon University, Seoul, Rep. of Korea
{hye_seung401, jasonlee88, hkim}@kw.ac.kr

ABSTRACT

Separating the leading singing voice from the musical background from a monaural recording is a challenging task that appears naturally in several music processing applications. Recently, kernel additive modeling with generalized spatial Wiener filtering (GW) was presented for music/voice separation. In this paper, an adaptive auditory filtering based on β -order minimum mean-square error spectral amplitude estimation (bSA) is applied to the kernel additive modeling for improving the singing voice separation performance from monaural music signal. The proposed algorithm is composed of five modules: short time Fourier transform, music/voice separation based on bSA, determination of back-fitting, back-fitting, and inverse short time Fourier transform. In the proposed method, the Singular Value Decomposition (SVD)-based factorized spectral amplitude exponent β for each kernel component is adaptively calculated for effective bSA-based auditory filtering performance during kernel back-fitting. Using a back-fitting threshold, the kernel back-fitting process can automatically be iteratively performed until convergence. Experimental results show that the proposed method achieves better separation performance than GW based on kernel additive modeling.

1. INTRODUCTION

A singing voice in a music signal contains useful information for a song, as it embeds the singer, the lyrics, and the emotion of the song. Therefore, vocal or singing voice separation from monaural music signal is an important task in many applications, such as automatic karaoke [1], instrument/vocalist identification [2], music/voice transcription, music remixing [3] and audio restoration.

So far, numerous vocal separation algorithms have been proposed with various approaches, such as non-negative matrix factorization [4], adaptive Bayesian modeling [5], and pitch-based interference [6-7]. These methods usually first map signals onto a feature space, then

detect singing voice segments, and finally apply source separation.

Recently, a relatively promising approach using kernel additive modeling (KAM) was proposed [8], wherein the spectrogram of each source is modeled only locally. This approach encompasses a large number of recently proposed methods for source separation [9-14]. KAM permits the use of different proximity kernels for different sources, with separation using an iterative kernel back-fitting (KBF) algorithm. In the kernel back-fitting, generalized Wiener filtering (GW) is used for the step of mixed music signal separation, and two-dimensional median filtering is applied to the power spectrogram of each source estimate for kernel spectrogram model fitting at each iteration. The GW requires good models of the spectrograms of each proximity source along with its spatial characteristics and permits very good separation provided these parameters are well estimated.

In spoken speech enhancement, one source may be the target voice, while others correspond to background noise which must be filtered out. Among the vast amount of single channel speech enhancement algorithms based on minimum mean-square error (MMSE) estimation of short-time spectral amplitude (STSA) published in the literature, it is well-known that the Bayesian STSA estimation methods [15] outperform the Wiener filtering, spectral-subtraction, and subspace approaches. In addition, among the Bayesian STSA estimation methods, β -order MMSE spectral amplitude estimation [15-17] achieved better enhancement performance than the existing Bayesian estimators, such as those based on the MMSE of the short-time spectral amplitude [15-17], and the MMSE of the logarithm of the STSA (LSA) [15-17].

In this paper, an advanced music/voice separation method is proposed, in which β -order MMSE spectral amplitude estimation and kernel spectrogram back-fitting are combined for improvement of the separation performance. In addition, the parameter β concerned in β -order MMSE spectral amplitude estimation is adaptively estimated according to the masking mechanism of human auditory system, the compressive nonlinearities of the cochlea and the critical sub-band SNR.

The proposed method has the following four advantages: (1) In the separation step, β -order MMSE estimation (bSA) of the factorized spectral amplitude



© Hye-Seung Cho, Jun-Yong Lee, Hyoung-Gook Kim.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Hye-Seung Cho, Jun-Yong Lee, Hyoung-Gook Kim. "Singing Voice Separation from Monaural Music Based on Kernel Back-fitting Using Beta-Order Spectral Amplitude Estimation", 16th International Society for Music Information Retrieval Conference, 2015.

was used instead of GW for the kernel back-fitting procedure to achieve better separation performances. (2) The Singular Value Decomposition (SVD)-based factorized spectral amplitude β_j were adaptively calculated for effective bSA estimation performance. (3) In the back-fitting step, an SVD-based factorization procedure was applied to the power spectrogram filtered by median filter to achieve efficient compression before processing of the next proximity source. (4) Using a back-fitting threshold, the kernel back-fitting process can automatically be iteratively performed until convergence.

This paper is organized as follows. Section 2 describes the proposed method, while Section 3 discusses the experimental results. Finally, the conclusion is presented in Section 4.

2. PROPOSED MUSIC/VOICE SEPARATION ALGORITHM

The proposed algorithm is composed of five modules: short time Fourier transform (STFT), music/voice separation based on β -order MMSE spectral amplitude estimation (bSA), determination of back-fitting, back-fitting, and inverse short time Fourier transform (ISTFT).

Figure 1 denotes the overall procedure of the proposed music/voice separation algorithm.

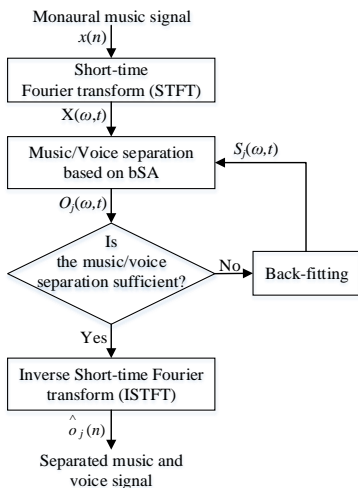


Figure 1. Overall flow chart of proposed music/voice separation algorithm.

We assume that the mixture music signal, $x(n)$, is taken as the sum of j underlying sources that are composed of some of percussive elements, one of the stable har-

monic elements, and one of the singing voice. Let a real-valued monaural music signal in discrete-time domain $x(n)$ be assumed as:

$$x(n) = \sum_{j=1}^J o_j(n) \quad (1)$$

where $j (= 1, 2, \dots, J)$ is index of each objective sources, n is sample index, and $o_j(n)$ denotes an objective source in mixture music signal.

First, an input monaural music signal $x(n)$ is transformed into the complex spectrogram $X(\omega, t)$ using the short-time discrete Fourier transform (STFT), as shown:

$$X(\omega, t) = \sum_{n=0}^{N-1} x(Rt+n)w(n)\exp\left(\frac{-j2\pi\omega n}{N}\right) \quad (2)$$

where R denotes the frame shift, t is the frame index, $w(n)$ indicates a window function, N is size of window, and ω is the frequency bin index, which is related to the normalized center frequency.

From the input complex spectrogram $X(\omega, t)$, complex spectrogram $O_j(\omega, t)$ for each objective sources is estimated by β -order MMSE spectral amplitude estimation.

Each current estimated spectrogram is compared with each previous estimated complex spectrogram. If the difference between the current and previous estimated spectrograms is not larger than the back-fitting threshold value, each complex spectrogram is converted back to the time domain using an inverse STFT. Conversely, if the difference between the two is larger than back-fitting threshold value, the kernel back-fitting process is iterated until convergence.

During the back-fitting processes, the power spectrogram of the estimated spectrogram is filtered by a simple two dimensional median filter with source-specific binary kernels. The source-specific binary kernels are explained in detail in next sub-section.

This kernel back-fitting proceeds in an iterative fashion, with alternate performance of separation and re-estimation (back-fitting) of the parameters to obtain new spectrogram estimates for each source.

2.1 Re-estimation using back-fitting

The re-estimation using back-fitting permits one to use different proximity kernels for each source and to separate them in order to perform the estimation. It assumes that vertical lines in a spectrogram correspond to percussive events; horizontal lines are typically associated with harmonics of pitched instruments, while cross-like forms correspond to singing voice events. In this case, peaks due to pitched harmonics can be regarded as outliers on the vertical lines associated with percussive events. Similarly, peaks due to the percussion events can be regarded as outliers on the horizontal lines associated with pitched harmonic instruments. Median filters used extensively in image processing are good at eliminating outliers. That is,

median filtering each time frame will suppress harmonics in this frame resulting in a percussion enhanced frame, while median filtering each frequency slice will suppress percussion events. This brings to the concept of using median filters individually in the horizontal, vertical, and cross-like directions to separate harmonic, percussive and vocal events.

The process is as follows:

(Step 1) Using the estimated complex spectrogram $O_j(\omega, t)$, the power spectrogram of the complex spectrogram is calculated as:

$$V_j(\omega, t) = |O_j(\omega, t)|^2 \quad (3)$$

(Step 2) A simple two dimensional median filter is applied to the power spectrogram $V_j(\omega, t)$ of the complex spectrogram with source-specific binary kernels, vocal, harmonic, and percussive. The different three proximity kernels [8] used for the median filter are as follows: (1) For a percussive and a repeating source, the vertical kernel is chosen; (2) For a harmonic source, the horizontal kernel is chosen; (3) Finally, for a source with only a spectral smoothness assumption, the cross-like kernel is chosen as vocals. The detailed three kernels are explained in the source separation using kernel additive models [8].

The median filtered kernel spectrogram is given by:

$$M_j(\omega, t) = \text{median}[V_j(\omega, t) | K_j(\omega, t)] \quad (4)$$

where $K_j(\omega, t)$ is a kernel which includes percussive elements of periodic components ($j = 1, 2, \dots, J-2$), the stable harmonic elements ($j = J-1$), and the singing voice ($j = J$), respectively. In effect, the original sample of the power spectrogram $V_j(\omega, t)$ of the complex spectrogram is replaced with the middle value obtained from a sorted list of the samples in neighborhoods of the original sample according to each kernel.

(Step 3) Kernel back-fitting using Wiener filtering or the β -order spectral amplitude estimator comes with an important drawback: it requires the full-resolution spectrogram, and storage of a huge amount of parameters in each iteration, and for each source. To reduce the memory usage and improve the separation performance while maintaining computational efficiency, Singular Value Decomposition (SVD) is applied to the full-resolution spectrogram $M_j(\omega, t)$:

$$S_j(\omega, t) = D_j \Sigma_j C_j = \text{SVD}[M_j(\omega, t)] \quad (5)$$

where $M_j(\omega, t)$ is factored into the matrix product of three matrices: the $M \times M$ row basis D_j matrix, the $M \times L$ diagonal singular value matrix Σ_j and the $L \times L$ transposed column basis functions C_j .

2.2 Separation using β -order MMSE spectral amplitude estimation

In the separation step, β -order MMSE spectral amplitude estimation of the factorized spectral amplitude is used

instead of GW for the kernel back-fitting procedure to achieve better music/voice separation performances. In the β -order MMSE spectral amplitude estimation, the spectral amplitude order β is quite important for singing voice enhancement or separation from monaural music signal. For the different β values, the gain values are different, and noise or other source reduction obtained is also different. In this way, the appropriated gain can be obtained by adaptively choosing right β .

However, the traditional calculation method about β is based on overall Signal-to-Noise Ratio (SNR) of each frame. That is, their values are fixed and not vary with frequency in each frame. Furthermore, the human auditory system has different sensitivity for different frequency components. Therefore, the b -th critical sub-band SNR is employed to calculate β values. For more effective bSA estimation performance, the Singular Value Decomposition (SVD)-based factorized spectral amplitude order $\beta_j(b, t)$ is adaptively calculated. Using adaptive β values and Singular Value Decomposition (SVD)-based factorized spectral amplitude, we can yield effective music/voice separation and obtain a good enhancement performance.

2.2.1 β -order MMSE spectral amplitude estimation

The β -order MMSE spectral amplitude estimation is composed of following four modules: sum of all $S_j(\omega, t)$, calculation of a priori SNR and a posteriori SNR, calculation of adaptive $\beta_j(b, t)$, and bSA-based gain function.

Figure 2 shows the β -order MMSE spectral amplitude estimation.

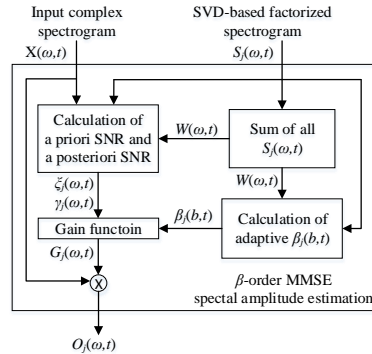


Figure 2. Overall flow chart of the β -order MMSE spectral amplitude estimation.

Before to obtain the estimated complex spectrum $O_j(\omega, t)$ from SVD-based factorized $S_j(\omega, t)$, the sum $W(\omega, t)$ of all $S_j(\omega, t)$ is defined by:

$$W(\omega, t) = S_1(\omega, t) + S_2(\omega, t) + \dots + S_J(\omega, t) \quad (6)$$

Then, the a priori SNR $\xi_j(\omega, t)$ and the a posteriori SNR $\gamma_j(\omega, t)$ of each objective proximity sources are calculated as follows:

$$\xi_j(\omega, t) = \frac{S_j(\omega, t)}{W(\omega, t) - S_j(\omega, t)}; \quad (7)$$

$$\gamma_j(\omega, t) = \frac{|X(\omega, t)|^2}{W(\omega, t) - S_j(\omega, t)}; \quad (8)$$

$$\chi_j(\omega, t) = \frac{\xi_j(\omega, t)}{1 + \xi_j(\omega, t)} \gamma_j(\omega, t); \quad (9)$$

where $\chi_j(\omega, t)$ is the function of $\xi_j(\omega, t)$ and $\gamma_j(\omega, t)$.

The gain function $G_j(\omega, t)$ for the bSA is given by:

$$G_j(\omega, t) = \frac{\sqrt{\chi_j(\omega, t)}}{\gamma_j(\omega, t)} \left[\Gamma \left(\frac{\beta_j(b, t)}{2} \right) \Phi \left(\frac{\beta_j(b, t)}{2}, 1; -\gamma_j(\omega, t) \right) \right]^{\frac{1}{\beta_j(b, t)}} \quad (10)$$

where $\Gamma(\bullet)$ is the gamma function, $\Phi(\bullet)$ is the confluent hypergeometric function. And $\beta_j(b, t)$ denotes the parameter based on the human auditory system.

To calculate $\beta_j(b, t)$, we employ the critical sub-band SNR. The b critical bands are divided for each speech frame, where a non-linear mel-frequency scale is used, which approximates the behavior of the auditory system. The mel-scale is a scale of pitches judged by listeners to be equal in distance one from another. The reference point between this scale and normal frequency measurement is defined by equating a 1000 Hz tone, 40 dB above the listener's threshold, with a pitch of 1000 mels. To convert a frequency ω in hertz into its equivalent in mel, the following formula is used:

$$pitch(mel) = 1127.0148 \log \left(1 + \frac{\omega(Hz)}{700} \right) \quad (11)$$

The spectrum is then processed by a mel-filter bank. The signal energy of the spectrum within b -th critical frequency sub-bands by means of a series of triangular filters whose center frequency are spaced according to the mel-scale. Thereafter, the critical sub-band SNR $Z_j(b, t)$ is calculated in the b -th band.

Finally, the estimated complex spectrogram from the gain function is defined as:

$$O_j(\omega, t) = G_j(\omega, t) \cdot X(\omega, t) \quad (12)$$

2.2.2 Calculation of adaptive $\beta_j(b, t)$

Since the spectral amplitude order $\beta_j(b, t)$ is based on characteristics of the human auditory system, including the compressive nonlinearities of the cochlea, and the perceived loudness, the choosing of adequate value for $\beta_j(b, t)$ can result in better enhancement or separation performance.

First, using $W(\omega, t)$ and $S_j(\omega, t)$, the sub-band SNR $Z_j(b, t)$ is calculated as:

$$Z_j(b, t) = 10 \log_{10} \frac{\sum_{\omega=B_{low}(b)}^{B_{up}(b)} |W(\omega, t) - \sqrt{W(\omega, t) - S_j(\omega, t)}|^2}{\sum_{\omega=B_{low}(b)}^{B_{up}(b)} (W(\omega, t) - S_j(\omega, t))} \quad (13)$$

where $b \in [0, 23]$ denotes the index of critical band. $B_{up}(b)$ and $B_{low}(b)$ denote the upper and lower frequency bound of the b -th critical band, respectively.

To obtain $\beta_j(b, t)$, the compression rate $\hat{\beta}_j(b, t)$ at intermediate frequencies can be calculated through linear interpolation between β_{low} and β_{high} . That is,

$$\hat{\beta}_j(b, t) = \beta_{high} - d(b, t)(\beta_{high} - \beta_{low}) \text{ for } 1 \leq j \leq J \quad (14)$$

using

$$d(b, t) = \frac{1}{B_{up}(b) - B_{low}(b)} \sum_{\omega=B_{low}(b)}^{B_{up}(b)} \left\{ \frac{1}{\eta} \log_{10} \left(\frac{f_{\omega} + l}{A} \right) \right\} \quad (15)$$

where $d(b, t)$ is the frequency-position function to the critical band, $\beta_{high} = 0.2$ and $\beta_{low} = 1$ denote the low-frequency and high-frequency of the compression rate, $\eta = 0.06$ mm, $l = 1$, and $A = 165.4$ Hz are the parameters set in paper [18], and f_{ω} is the frequency in Hz corresponding to spectral component ω , i.e., $f_{\omega} = \omega F_s / N$, where F_s is the sampling frequency.

By limiting the range of $\check{\beta}_j(b, t)$ as $[\beta_{min}, \beta_{max}]$ in order to obtain a better trade-off between target source enhancement and other source reduction, $\check{\beta}_j(b, t)$ can be calculated through the following relationship:

$$\check{\beta}_j(b, t) = \min \{ \max \{ \mu \cdot Z_j(b, t) + \lambda, \beta_{min} \}, \beta_{max} \} \quad (16)$$

where $\mu = 0.45$, $\lambda = 1.3$, $\beta_{min} = 0.4$, and $\beta_{max} = 4.0$.

According to sub-band SNR, the compressive nonlinearities of the cochlea, and perceived loudness, a parameter $\beta_j(b, t)$ is given as follows:

$$\beta_j(b, t) = q \cdot \hat{\beta}_j(b, t) + (1-q) \cdot \check{\beta}_j(b, t) \quad (17)$$

where q ($0 < q < 1$) is a smoothing parameter.

3. EXPERIMENTAL RESULTS

In this section, the performance of the proposed bSA-KBF algorithm is evaluated for the separation of background music and singing voice.

For experiments, 100 full-length song tracks were used (50 songs from the ccMixer database containing many different musical genres, 50 songs from a self-recording studio music database), where all singing voices and music accompaniments were recorded separately. All of the song data were stored in PCM format with mono, 16-bit depth, and 44.1 kHz sampling rate.

For each track, the accompaniment of 6 repeating patterns along with a 2 second steady harmonic source was determined. Vocals were modeled using a cross-like ker-

nel with a height of 15 Hz and width of 20 ms. The frame length was set to 90 ms, with 80% overlap. Six to eight iterations were performed for the back-fitting algorithm (approximately until convergence).

For the performance measures, performance was evaluated in terms of Normalized Source-to-Interference Ratio (NSIR) and Normalized Source-to-Distortion Ratio (NSDR) by Blind Source Separation Evaluation (BSS Eval) metrics [19]. NSDR and NSIR for singing voice are defined as:

$$\begin{aligned} \text{NSDR}(v_r, v, x) &= \text{SDR}(v_r, v) - \text{SDR}(x, v) \\ \text{NSIR}(v_r, v, x) &= \text{SIR}(v_r, v) - \text{SIR}(x, v) \end{aligned} \quad (18)$$

where v_r is the synthesized singing voice, v is the original clean singing voice, and x is the mixture. NSDR is for estimating the improvement of the SDR between the processed mixture x and the separated singing voice v_r . Higher values indicate better separation.

The performance of the proposed bSA algorithm was compared with those of GW, LSA based on KAM.

Table 1 presents the experimental results of comparative performance for music/voice separation of the four methods:

- STFT-GW-KAM: As a basic KAM algorithm, the generalized Wiener filter was applied to the power spectrogram based on STFT.
- SVD-GW-KAM: SVD was performed on the power spectrogram based on STFT. To the SVD-based decomposed power spectrogram, the generalized Wiener filter was applied.
- SVD-LSA-KAM: The MMSE of the logarithm of the STSA was applied to the SVD-based decomposed power spectrogram.
- SVD-bSA-KAM: β -order MMSE STSA was applied to the SVD-based decomposed power spectrogram.

Methods	Separation Performance for Music		Separation Performance for Voice	
	NSDR	NSIR	NSDR	NSIR
STFT-GW-KAM	6.37	9.18	1.89	5.76
SVD-GW-KAM	6.83	9.65	2.35	6.23
SVD-LSA-KAM	7.36	10.48	2.87	6.74
SVD-bSA-KAM	8.25	12.13	3.12	6.88

Table 1. Comparative performance for music/voice separation.

As shown in Table 1, the best separation performance of the music from the mixed music signal is obtained with the proposed method, SVD-bSA-KAM, in terms of NSDR and NSIR. Compared to the other three methods, the basic method, STFT-GW-KAM, attains the worst results. And the proposed bSA delivers high performance result in the separation of vocal components.

4. CONCLUSIONS

In this paper, we proposed a β -order MMSE spectral amplitude estimation method based on kernel back-fitting for music/voice separation. The proposed algorithm enhances the basic kernel back-fitting algorithm through application of β -order MMSE spectral amplitude estimation considering the perceptual properties of human auditory system. The experimental results show that the proposed method obtained better results compared to other existing methods.

In future work, we will apply the method to spatial audio reproduction applications running on smart phones.

5. ACKNOWLEDGMENTS

This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education, Science and Technology(NRF-2013R1A1A2007601). And this research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2015-H8501-15-1016) supervised by the IITP(Institute for Information & communications Technology Promotion)

6. REFERENCES

- [1] Z. Rafii and B. Pardo: "Repeating Pattern Extraction Technique (REPET): A Simple Method for Music/Voice Separation," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 21, No. 1, pp. 73–84, 2012.
- [2] N. C. Maddage, C. Xu, and Y. Wang: "Singer identification Based on Vocal and Instrumental Models," in *Proceedings of the 17th International Conference on Pattern Recognition*, Vol. 2, pp. 375–378, 2004.
- [3] S. Marchand et al: "DRaM: A Novel System for Joint Source Separation and Multi-Track Coding," in *Proceedings of the 133rd Audio Engineering Society Convention*, 2012.
- [4] S. Vembu and S. Baumann: "Separation of Vocals from Polyphonic Audio Recordings," in *Proceedings of the International Society for Music Information Retrieval Conference*, pp. 337–344, 2005.
- [5] A. Ozerov, P. Philippe, F. Bimbot, and R. Gribonval: "Adaptation of Bayesian Models for Single-Channel Source Separation and its Application to Voice/Music Separation in Popular Songs," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 15, No. 5, pp. 1564–1578, 2007.

- [6] Y. Li and D. Wang: "Separation of Singing Voice From Music Accompaniment for Monaural Recordings," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 15, No. 4, pp. 1475–1487, 2007.
- [7] C. -L. Hsu and J. -S. R. Jang: "On the Improvement of Singing Voice Separation for Monaural Recordings Using the MIR-1K Dataset," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 18, No. 2, pp. 310–319, 2009.
- [8] A. Liutkus, D. Fitzgerald, Z. Rafii, B. Pardo, and L. Daudet: "Kernel Additive Models for Source Separation," *IEEE Transactions on Signal Processing*, Vol. 62, No. 16, pp. 4298–4310, 2014.
- [9] Z. Rafii and B. Pardo: "Repeating Pattern Extraction Technique (REPET): A Simple Method for Music/Voice Separation," *IEEE Transaction on Audio, Speech and Language Processing*, Vol. 21, No. 1, pp. 73–84, 2013.
- [10] D. Fitzgerald: "Harmonic/Percussive Separation Using Median Filtering," in *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx-10)*, 2010.
- [11] Z. Rafii and B. Pardo: "A Simple Music/Voice Separation Method Based on the Extraction of the Repeating Musical Structure," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 221–224, 2011.
- [12] A. Liutkus et al: "Adaptive Filtering for Music/Voice Separation Exploiting the Repeating Musical Structure," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 53–56, 2012.
- [13] Z. Rafii and B. Pardo: "Music/Voice Separation Using the Similarity Matrix," in *Proceedings of the International Society for Music Information Retrieval Conference*, pp. 583–588, 2012.
- [14] O. Yilmaz and S. Rickard: "Blind Separation of Speech Mixtures via Time-Frequency Masking," *IEEE Transaction on Signal Processing*, Vol. 52, No. 7, pp. 1830–1847, 2004.
- [15] E. Plourde and B. Champagne: "Auditory-Based Spectral Amplitude Estimators for Speech Enhancement," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 16, No. 8, pp. 1614–1623, 2008.
- [16] F. Deng, F. Bao and C. -C. Bao: "Speech Enhancement Using Generalized β -Order Spectral Amplitude Estimator," in *Proceedings of the Speech Communication*, Vol. 59, pp. 55–68, 2014.
- [17] C. H. You, S. N. Koh, and S. Rahardja: " β -Order MMSE Spectral Amplitude Estimation for Speech Enhancement," *IEEE Transactions on Speech and Audio Processing*, Vol. 13, No. 4, pp. 475–486, 2005.
- [18] D. D. Greenwood: "A Cochlear Frequency-Position Function for Several Species-29 Years Later," *Journal of Acoustic Society America*, Vol. 87, No. 6, pp. 2592–2605, 1990.
- [19] E. Vincent, R. Gribonval, and C. Fevotte: "Performance Measurement in Blind Audio Source Separation," *IEEE Transactions on Audio, Speech and Language Processing*, Vol. 14, No. 4, pp. 1462–1469, 2006.

SCHEMATIZING THE TREATMENT OF DISSONANCE IN 16TH-CENTURY COUNTERPOINT

Andie Sigler

School of Computer Science,
McGill University;
Computing Music

Jon Wild

Schulich School of Music,
McGill University
wild@music.mcgill.ca

Eliot Handelman
Computing Music

ABSTRACT

We describe a computational project concerning labeling of *dissonance treatments* – schematic descriptions of the uses of dissonances. We use automatic score annotation and database methods to develop schemata for a large corpus of 16th-century polyphonic music. We then apply structural techniques to investigate coincidence of schemata, and to extrapolate from found structures to unused possibilities.

1. INTRODUCTION

We develop a set of schematic dissonance treatments (i.e. schemata under which the uses of dissonance are classifiable) using a large corpus of mass movements (almost 1000) of Palestrina and Victoria, dating from the 16th century. Palestrina in particular has a resonance through the history of music as one whose style was raised to the status of a didactic norm.¹ As a result, Palestrina’s practice (or a simplification of it) has been well known and imitated for centuries among academics and music students.² As a foil for Palestrina, we compare masses by Victoria, roughly contemporaneous and with a similar dissonance treatment. The wealth of available literature on the dissonance practice of this style gives us a departure point for developing a computational platform for its investigation, with a view to generalization.

¹ As pointed out in Alfred Mann’s 1991 forward to Jeppesen’s *Counterpoint* [2], one of several classic texts on the Palestrina style – as the title shows, the name Palestrina is all but synonymous with certain aspects of basic musical organization – in particular the way a “point” (i.e. a note – or perhaps a musical “idea”) sounds and moves “counter” to (i.e. in relation to) another point or set of points.

² Including e.g. Haydn, Mozart, Beethoven, Schubert, Rossini, Chopin, Berlioz, Liszt, Brahms, Bruckner, R. Strauss, and Hindemith, who all are known to have used Fux’s *Gradus ad Parnassum*, based on the Palestrina style ([1], Mann’s introduction).



© Andie Sigler, Jon Wild, Eliot Handelman.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Andie Sigler, Jon Wild, Eliot Handelman. “Schematizing the Treatment of Dissonance in 16th-Century Counterpoint”, 16th International Society for Music Information Retrieval Conference, 2015.

2. METHODOLOGY

2.1 Automatic Score Annotation

Part of our methodology for investigating dissonances is to look at *automatically annotated scores*. Seeing annotated scores helps us evaluate the correspondence of our specification to our intention, and develop to new schemata. It allows us to identify musical factors that would likely not have been apparent otherwise (i.e. in a situation where data was displayed in a musically non-intuitive way, or where scores had to be painstakingly scrutinized to locate scarce occurrences).

Using a web-based music analysis system produced by Computing Music, we generate annotated scores *on-demand* (at load time). It’s possible to load and analyse any score (including ones outside the corpus under investigation), to load a random score from the corpus, or to “spin” through a corpus with a search for instances of a particular configuration, such that one keystroke displays a new annotated score focussed on the relevant measure, bringing together similar occurrences from disparate locations.

2.2 Saving Features

On a first pass through the score, we save a set of *features* for each dissonance, including duration, surrounding melodic intervals, metric weight, and type of attack, as needed to define our schemata (– as we developed and added new schemata, an initial set of features was expanded). Any features not used in a given schema are open to any value.

Saving a set of features for each dissonance rather than just applying a set of schematic filters on the first pass through the score has certain advantages. Suppose we run all dissonances through a set of filters, and several of them are labelled P for passing. Now if we want to ask questions about the set of passing notes (in fact matched by several different but related schemas) – e.g. how many are going up or down, or how many are half-notes – we have to *re-ask* some of the same questions we already asked in order to label them in the first place. As well, if we have a set of remaining *unlabelled* dissonances, we will have no idea *how* they failed the tests for the different labels, or what subsets of unlabelled dissonances might have in common.

We save feature-sets (and schema labels) in a database,

so that we can query them in different ways; database searching also helps us develop schemata based on featural similarities of unschematized dissonances.

3. DEFINITIONS AND SCHEMATA

3.1 Dissonance and Meter

As in standard practice, we define the *dissonant intervals* as the minor and major second, perfect fourth, tritone, and minor and major seventh, and their compounds (i.e. with additional octaves). A *dissonance* occurs when two notes coincide or overlap in time and form a dissonant interval.

From the initial set of dissonances in a score, we remove certain fourths and tritones that participate in sonorities considered consonant.³ If a perfect fourth is accompanied by an additional voice sounding a third or fifth (or their octave compound) below its lower note, it is considered consonant. Likewise a diminished fifth accompanied by the pitch a major sixth below its lower note, or an augmented fourth accompanied by the pitch a minor third below its lower note.

Meter can be thought of as a temporal grid. We generalize to metric *weight*, where different places in the measure are said to be equally “strong” or “weak.” The downbeat is the strongest, followed by the divisions into halves, then divisions into quarters, then eighths.

The meters under consideration are *duple*, using whole-note divisions (e.g. 4/2; 2/2), or *triple*, using dotted-whole divisions (e.g. 3/2, 6/2, 9/2). We don’t differentiate between whole notes in a duple meter or dotted wholes in a triple meter; each one represents an equal beat.

3.2 One- and Two-voice Schemata

Although a dissonance is defined as a relationship between two notes in two different voices, commonly only one note needs to be “explained.”⁴ Typically, when one note is struck and then sustained (or reattacked) while the second note is struck (an *oblique* motion), and the second note is on a “weak” metric position, only the *second* note needs explanation, since the dissonance only occurs once the second note enters – we call the second note the dissonance (with respect to the first note). In these kinds of cases, we can schematize a dissonance treatment with respect to features of the voice containing the dissonance, and not the voice against which it dissonates. For example, a *passing* note is schematized by either of two different melodic shapes: (step up, step up) or (step down, step down).⁵ It is simultaneously schematized by one of four different metric shapes: a half note on a weak half preceded by a duration of at least a half, a quarter note on a weak quarter, an eighth on a weak quarter, or an eighth on a weak eighth. We

³ These correspond to major and minor triads in root position or first inversion, and diminished triads in first inversion, though these designations are anachronistic for the 16th century.

⁴ Informally, *explaining* means locating a theorized schematic dissonance treatment to which a dissonance corresponds.

⁵ I.e. (step up, step up) gives a figure of *three notes*, including a step up to the dissonating note called the “passing note”, and another step up from the passing note.

have defined several other “single-voice” schemata; these are summarized in Table 1.

A *suspension* is a *two-voice* schema, involving the suspended note as well as its counterpart, the “agent.” The agent, or active voice, is an obliquely struck dissonance on a strong beat, after which the *other* voice (the suspension) is constrained to resolve downward by step. Since we’ve already set up machinery to find *attacked* dissonances, rather than to find notes that are dissonated against at a particular place in their duration, it’s convenient to start the schematization of suspensions with the agent, rather than with the suspension note itself. When we find an oblique dissonance on a strong beat, we can pull in a feature set for the note against which it dissonates, and check whether the combination constitutes a suspension. Further description of suspensions can be found in Table 2.

3.3 Extending the pairwise model

We originally defined dissonances as occurring between *two* voices. One exception to this model that we have already addressed is the consideration as consonant of fourths and tritones that are covered by certain notes in a lower-sounding voice – these are “vertical” or *harmonic* schemata. Apart from these, we have so far used a pairwise model to schematize dissonances between any two voices. But we found we had to extend the pairwise model to account for some dissonances. These are summarized in Table 3.

1. We find that if a note is consonant with an agent of a suspension, it can be dissonant with the suspension without further constraint; as well, we find situations where a note is dissonant with an agent, but explicable as consonant with the suspension.

2. On a weak quarter, two quarter notes or eighths (or one of each) may be dissonant with respect to each other, if there is a third voice such that each is explained as consonant, passing, neighboring, a *cambiata*, or an anticipation with respect to the (same) third voice. (See Figure 1.)

3. A note *m* that is dissonant within a given pair of voices is in condition M if it has the same pitch class as a note in a third voice that was already sounding when *m* entered, and is sustained at least until the end of *m*. Notes in condition M are often approached and left by leap. A note in condition M may be attacked simultaneously with a dissonance; in this case the note not labelled M will be explained (e.g. as a passing or neighbor note) with respect to the third voice.⁶

4. DISCUSSION: EXCEPTIONS AND INDUCTION

At the time of this writing, there are still ~360 dissonances in the Palestrina-Victoria corpus that are not explained by

⁶ In fact, if we look at half notes that are dissonant counterparts to condition M, we find that they are *all* passing notes, with six exceptions that are upper neighbors – and these six are all in the same mass of Victoria. This is an example of a unique dissonance treatment, used motivically, that is clearly related to the more common passing version.

Symbol	Name	Melodic schema	Metric schema	Attack
P	Passing	(step up, step up) (step dn, step dn)	weak quarter eighth on weak quarter weak eighth weak half after \geq half	oblique
N	Neighbor	(step up, step dn) (step dn, step up)	(same as for P)	oblique
C5,C4,C3	(5/4/3-note) Cambiata	(step dn, third dn, step up, step up) – or first n notes of this	weak quarter	oblique
A	Anticipation	(step, repeat)	weak quarter weak eighth	oblique
E	Echappée	(step up, leap dn) (step dn, leap up)	weak quarter	oblique
F	“Fake” suspension	(step or repeat, step dn)	syncopated whole syncopated half syncopated dotted-half	oblique
Q,Qx	Third quarter	(step dn, step dn)	quarter on weak half; Q if after \geq half, otherwise Qx	oblique simultaneous
L	Leap of third	(third dn, step up)	weak quarter	oblique

Table 1: “Single-voice” Schemata

Symbol	Name	Description
S,G	Suspension, Agent	Suspension S is sustained or reattacked on the same note; agent G strikes oblique dissonance; S moves down (to its <i>resolution</i>) by step on a weaker beat than G.
T,T2,G	Suspension with third-skip, Agent	As S, but with resolution (third dn, step up) in quarter notes; the note skipped down to can be dissonant (called T2).

Table 2: Suspensions

Symbol	Name	Description
Gc	Consonant with Agent	Dissonant with a suspension S or T but consonant with its agent. Or dissonant with a “Fake” suspension F and consonant with its “agent.”
Sc	Cons. with Suspension	Dissonant with an agent but consonant with its suspension.
M/M2, Mx	Match	Has the same pitch / pitch class as a note in a <i>third</i> voice already sounding when M entered, and is sustained at least until the end of M. M’s dissonant counterpart Mx is attacked simultaneously with M; explained as P or N with the <i>third</i> voice.
W	Weak-quarter clash	On a weak quarter, a dissonance between two quarter notes or eighths (or one of each), such that each is consonant, passing, neighboring, a cambiata, or an anticipation with respect to some (same) third voice.

Table 3: Schemata: Extending the Pairwise Model

any of our schemata (versus ~194100 that *are* – we’ve successfully schematized > 99.8% of dissonances in the corpus). Unschematized dissonances are marked with an X in annotated scores. There are quite a few errors (e.g. wrong notes or durations) in our corpus, and it looks like a considerable proportion of Xs are due to these. The ability to quickly navigate to problematic dissonances allows us to make corrections where they are necessary (i.e. by comparison with another edition) – correction of the corpus is currently underway. This method doesn’t locate *all* errors in the corpus, but it does point out especially “bad” ones from the point of view of dissonance.

Examining Xs is also part of our development methodology for formulating new dissonance categories. For instance, by doing some filtering on a database of unmatched feature-sets, we noticed that there were 54 unschematized

dissonances that are on weak quarters and are approached by a third down and left by a step up. We wrote this schema into our specification (“L” in Table 1), and then were able to “spin” through the instances in the corpus to see whether the schema met our expectations on the annotated scores.⁷

In another database exploration case, we began by observing that there were quite a few unschematized dissonant half- and quarter-notes on beat one, which were approached and left by a step. This preliminary schemati-

⁷ After finding this dissonance in the database, we observed that it is mentioned (as possibly an “archaism”) in [3], p. 220. Jeppesen’s study proves to be a tour de force of detail – for example, on p. 268 he shows a suspension that jumps down a fifth before leaping back up a fourth to its resolution, saying that as far as he’s observed, “this occurs but once in the whole collection of Palestrina’s compositions,” despite being a standard practice in [1]. We don’t find a second occurrence in Palestrina, and it occurs once in our Victoria corpus.

Figure 1: W (weak-quarter clash) are “explained” dissonances (or consonances) with respect to other voices (in this case P and CS), but simultaneous, unclassified dissonances with respect to each other.

Figure 2: The tied note in the third voice marked “F S” is a fake suspension (F) with respect to the bottom voice, and a “real” suspension (S) with respect to the agent (marked G) in the voice above.

zation was obviously too general to keep as a final labeling, since we don’t wish to allow passing and neighboring notes on strong beats indiscriminately. But looking through these instances showed us that (along with a small number of less explicable occurrences), there were a couple of schematic situations. One such situation occurred when the dissonance in question had an *agent* as its dissonant counterpart, while being *consonant* with the corresponding *suspension* (shown as label Sc in Table 3). We also were able to refine our definition of *suspensions* by looking at these unschematized strong-beat dissonances. Our original definition stipulated that the agent must be *consonant* with suspension’s note of *resolution* (whether or not the agent is still sounding at the time of the resolution). In fact, we find there is one situation where this rule doesn’t hold: when the suspended interval is a diminished fifth, resolution forms a fourth (– dissonant) with the

Figure 3: A unique structure of simultaneous dissonances: two passing notes, a neighbor, and a cambiata. (Palestrina: Laudata Dominum, Gloria)

agent. When this happens, the agent always moves up a step to meet the resolution in a (consonant) *third*.

When exploring for new schemata, we sometimes come against occurrences that are interestingly rare. For instance, we find that there are six third-quarter passing notes going *upward* in the combined Palestrina-Victoria corpus. Of these, four are in one mass of Victoria, and are essentially repetitions of the same single situation. The remaining two are separate instances in Palestrina. This kind of instances open musicological questions as to the interpretation of these scarce occurrences: *why* was this possibility used just here, and practically nowhere else.

Database exploration works not only for induction of new schemas, but for deeper exploration of defined schemas. For instance, if we look at the feature set for the relatively rare half-note lower-neighbors, we find that *most* of them (in Palestrina 119/150, or 79%) are a perfect fourth above the note they dissonate with. A few (13, or 9%) are a tritone below, and on closer inspection, these all seem to take part in very similar cadential figures. Victoria uses the tritone/cadential lower neighbor somewhat more often – 20/83 or 24%, and the perfect fourth above 43/83 or 52%: a similar but less dramatic tendency.

Likewise, we find that our category for “fake” suspension (F) (which Jeppen calls a “consonant fourth”) never occurs with a tritone, and in fact *always* occurs with either a fourth, or (less frequently) a fourth *and* seventh or second (i.e. with respect to two different voices). Furthermore, the F which is *only* a fourth at its onset is almost always accompanied by a suspension (S) of a seventh or second on the next strong beat (Figure 2) – the fake suspension of

a fourth with *no* seventh or second at all is found only 16 times in the Palestrina corpus and never in Victoria. We could continue this line of musicological investigation by surveying for further details, finding e.g. those fake suspensions which are a half note in duration, or those introduced by leap, or those which include a dissonance of a *minor* second or *major* seventh (a rare occurrence), or those which have a resolution of a *major* second (relatively rare).

We wonder: would it be feasible to *automatically* induce dissonance treatments over a corpus (i.e. start from scratch and have a search deliver a set of schemata that are used a minimum number of times in a corpus). Although this would be computationally expensive, it seems possible.

The strategy for doing so, however, is not completely transparent. If we address the subset of one-voice schemata, we can imagine trying to cover the set of dissonances with minimal explanatory schemata (with the heuristic that more proximate intervals have to be part of a schema before more distant intervals can be included). For conjunctions, this is straightforward enough (e.g. must be on a weak quarter *and* resolve down). For disjunctions, we would have to infer whether a reduced set of features should be specified, or whether to use a wild card. We would have to be careful not to overfit schemata, which would result in a large number of highly specific schemata instead of a smaller number of more general ones (e.g. a passing note figure, once completed can be followed by a step up, or a leap up, or a third down, etc.). There's also no obvious way of joining multiple discovered schemata under one descriptive tag. For example, eight different schemata emerge for what we call "passing notes" (depending on their position, duration, and orientation) – and this is not including third-quarter passing notes, which we have chosen to name differently.

The schemata found would be constrained to be described by the feature set we're examining. We've used shorthand features such as "weak quarter," generalizing second and fourth quarters, and "leap up," generalizing several intervals. If we started off an automatic schema induction with these generalized features, it would be powerless to differentiate them (– generalizing *reduces* our power as human experts to differentiate them, but we still stand a chance of doing so by looking at scores). On the other hand, if we start with a *larger* feature set, we increase the search space exponentially, but add an interesting layer of *feature* induction. Even if we start with a larger feature set, we're still constrained by pre-process feature selection, whereas humans are free to add features midstream.

We won't discuss here the added problem of trying to induce two-voice schemata such as suspensions from scratch, nor the various three-voice schemata. We would also need to consider *harmonic* treatment: dissonances may be treated differently when they're a part of a chord (aside from the chords we have already discussed, for some corpora seventh chords, root-position diminished triads, or second-inversion triads have special status). Having errors in the corpus also complicates the picture.

While automatic schema induction is an interesting concept, for the time being it seems that using database queries and automatic score annotation to facilitate deep interaction of human intelligence with a musical corpus is still the most effective procedure.

5. STRUCTURING DISSONANCES

So far, what we've described are specific filters defined on feature vectors. These filters assign tags to notes, labeling the dissonance treatment of the note. Now we have the opportunity to see how these dissonance treatments interact. For instance, it's quite common to have two or more passing notes in different voices at the same time. What other combinations of dissonance might occur? For this analysis, we don't have to develop new schemata and filter for them, we merely have to *build structures* out of the dissonances we already have.

The procedure is this: we take a set of labeled dissonances, and build graphs of temporal relations between them. For the purpose of this example, we keep the space small by only examining a subspace of temporal relations between dissonances. We use three types of temporal relation: monophony (i.e. one or more notes beginning and ending at the same time), inclusion (i.e. a note's duration being *within* the duration of another note), and overlap.

We also use a subset of dissonances: passing and neighboring tones, third-quarters, anticipations, échappées, cambiatas, dissonant leaps of a third, "real" and "fake" suspensions, and weak-quarter clashes. The experiment reduces each score to *just* the notes marked with these labels, and then constructs polyphonic structures out of the remaining subscore. That is, we will connect tagged dissonances that are in temporal contact with one another, then examine sets of connected dissonances in the corpus. In what follows, we are counting *not notes*, but *structures*, which can contain one or more notes.

We obtain 297 different structures by this method – 243 in Palestrina and 175 in Victoria, with 121 in their intersection, and therefore 122 in Palestrina but not Victoria and 54 in Victoria but not Palestrina.⁸ Of the 297, 113 occur only *once* in the combined corpus, while another 80 occur fewer than five times. In general, we see a relatively small number of structures occurring very frequently, and a large number of structures occurring rarely.

The most common structures in Palestrina and Victoria differ only slightly. The most frequent for both composers is the lone passing note, followed by the suspension, the double-passing note (i.e. two simultaneous passing notes), and then the (lone) neighbor. The next most common for Palestrina is the third-quarter passing note, then simultaneous passing and neighbor notes, and simultaneous neighbor notes. Victoria would be the same, except the third

⁸ The absolute numbers themselves are not of great interest, and we don't offer a proper statistical analysis, we only mean to give a general orientation as to the structural variety available from the point of view of this experiment. The numbers are, furthermore, provisional since we're still correcting the corpus, but the great majority of rare structures are *not* due to corpus errors.

quarter dissonance is slightly more rare in Victoria, appearing after the latter two.

Other structures show a greater difference in practice between the composers. For instance, just looking at structures with double suspensions, we find that Palestrina resolves these at different times (i.e. one resolution coming a quarter note before the other) over half of the time, whereas Victoria only resolves them at different times about 10 percent of the time.⁹ We find also that simultaneous “fake” suspensions don’t occur in Palestrina, while there are 26 instances in the Victoria corpus. A figure in which a note is a dissonant third-quarter with respect to one voice at the same time as being interpreted as the agent of a diminished suspension¹⁰ in another voice is found 23 times in Palestrina and once in Victoria. The cambiata occurring within the duration of a suspension, and the double third-quarter dissonance are also much more frequent in Palestrina than in Victoria. Everything found in Victoria more than three times is also found in Palestrina at least once – it’s not obvious if this is an artifact of the difference in the sizes of the corpora,¹¹ or whether it reflects on the practice of the composers.

6. EXTRAPOLATION AND NEW STRUCTURES

The distribution of structures of labelled dissonances, with many structures used only once or a handful of times, shows us that we are not dealing with a closed set of reusable possibilities, but a *composable* space. This suggests that it’s possible to build structures that are not in the corpus, but that are within the matrix of possibilities outlined by the corpus. In efforts to build style-copying automata, a trend has been to *re-use* and *re-combine* elements found in a corpus. But since it is the responsibility of the artist to offer *something new* in each work, reasoning about *unused* structures is essential for deeper exploration of corpus extension.

What we present in this section was not constructed automatically; we simply show that the structures we obtained from labelled dissonances seem to constitute a set with missing elements which *might* have been used in the corpus. It is our opinion that it would be possible to construct these automatically, and that in any case, the set of unused possibilities (and the set of once-used possibilities) are an avenue of insight into the nature of composition. Our ability to schematize the treatment of a great majority of dissonances in the corpus points to a constrained and rule-bound composition practice. How does this relate to the obligation to create new and different works? And is it possible to *reason* about newness and difference? In this section we suggest an approach.

We can proceed rather conservatively: instead of trying to invent complex and exotic new combinations that might be realizable, we can start by looking for unfilled

niches that are relatively simple. For instance, if we take the subset of structures that consist of more than two *simultaneous* dissonances including at least one cambiata and one neighbor, we find a *single structure* that occurs once: a cambiata, neighbor, and two passing tones at the same time. This means that the *simpler* cambiata, neighbor, and *one* passing tone never occurs! We also see other obvious combinations including a cambiata and two neighbors, two cambiatas and a neighbor, and a cambiata, two neighbors and one passing note. It is simple to enumerate all of the possibilities in this small combinatorial space.¹²

For a given constructed dissonance structure, it’s not guaranteed that it is realizable. We can try to realize it systematically by generating and testing candidates. The space of candidates is small enough to be tractably enumerable, especially if we proceed in stages, leaving the issue of voicing (order of voices from low to high) until later. Candidates can be rejected if they cause unschematized dissonances (Xs), or break some other constraint – e.g. we might reject parallel fifths, octaves, and unisons, to conform with the style. It turns out that we can construct viable fragments in which a cambiata, a neighbor, and a passing tone occur simultaneously, or in which a cambiata and two neighbors occur simultaneously (left as an exercise for the reader!). As far as we can tell, there’s no “reason” that these don’t occur in the corpus.

We can extend this game of finding unused potentials by taking the interval combinations of a structure as another parameter. For instance, four simultaneous passing notes occur about 40 times in the combined corpus, but most of the time the passing note “chord” is just a minor or major third, with pairs of passing notes up and down through each note of the third. There is one instance where a minor triad is constituted (one passing note is preceded by a dissonant third quarter). The major triad occurs several times in the triple-passing-tone structure; it appears to be an unused possibility in the quadruple.

The possibility for combinatorial explorations are vast. For instance, there are more than 70 different *sonorities* (pitch-class sets sounding at some moment) in Palestrina, while only 7 of them need not involve dissonance. The rest are constructed precisely in the manner we have just been describing, with combinations of dissonance treatments.

Equally great are the opportunities for musicologists to study specific usages in their musical, textual, and historical contexts; the computational means to find and annotate sets of occurrences will surely facilitate this process.

The general methodology used here can be extended to other corpora, and to other aspects of musical practice. The computational study of musical corpora through schematization, structure-building, automatic annotation, and generative extrapolation will bring a new scope and precision to our understanding of musical practice and potential.

⁹ We can see this because these two instances have different polyphonic profiles: if they both resolve at the same time they’re in rhythmic monophony with one another, whereas if one resolves first, one suspension is durationally contained within the other.

¹⁰ I.e. a suspension with duration of a quarter.

¹¹ 261 movements of Victoria vs. 705 of Palestrina

¹² In fact the *whole* space of dissonance structures under this model may be small enough to be feasibly enumerable. If so, how does this fact relate to our surmise that *for Palestrina and Victoria*, the space seems to be “composed” rather than enumerated? This is a question for the practice and philosophy of the nascent discipline of constructive musicology, or the study of corpora through computational extension.

7. REFERENCES

- [1] Fux, J.J.; Mann A. (trans. & ed.): *The Study of Counterpoint from J.J. Fux's Gradus Ad Parnassum*. Norton & Co. (1971/1965/1725)
- [2] Jeppesen, K.: *Counterpoint: The Polyphonic Vocal Style of the Sixteenth Century*. Dover Publications (1992/1939/1931)
- [3] Jeppesen, K.: *The Style of Palestrina and the Dissonance* Dover Publications (1970/1946)
- [Sch.1999] Schubert, P.: *Modal Counterpoint, Renaissance Style* Oxford University Press (1999)

PREDICTIVE POWER OF PERSONALITY ON MUSIC-GENRE EXCLUSIVITY

Jotthi Bansal

McMaster University
bansal.j@mcmaster.ca

Matthew Woolhouse

McMaster University
woolhouse@mcmaster.ca

ABSTRACT

Studies reveal a strong relationship between personality and preferred musical genre. Our study explored this relationship using a new methodology: genre dispersion among people's mobile-phone music collections. By analyzing the download behaviours of genre-defined user subgroups, we investigated the following questions: (1) do genre-preferring subgroups show distinct patterns of genre consumption and genre exclusivity; (2) does genre exclusivity relate to Big Five personality factors? We hypothesized that genre-preferring subgroups would vary in genre exclusivity, and that their degree of exclusivity would be linearly associated with the openness personality factor (if people have open personalities, they should be "open" to different musical styles). Consistent with our hypothesis, results showed that greater genre inclusivity, i.e. many genres in people's music collections, positively associated with openness and (unexpectedly) agreeableness, suggesting that individuals with high openness and agreeableness have wider musical tastes than those with low openness and agreeableness. Our study corroborated previous research linking genre preference and personality, and revealed, in a novel way, the predictive power of personality on music-consumption.

1. INTRODUCTION

Existing music-personality studies have specifically examines the relationship between music preference and Big Five personality factors [4, 13, 16]. The music people listen to—their musical preferences—reveal aspects of their identity [12], to the point where music can be worn as a "badge" of honour [16].

Big Five personality factors are designed to delineate basic, measurable features of personality. Each factor consists of various traits that describe behaviour, thoughts and emotions; traits that co-vary with one-another are categorized under one factor [3]. Factors in the current Big Five model are openness, conscientiousness, extraversion, agreeableness, and neuroticism. Each factor is defined based on terms from everyday language [7].

In detail, the Big Five personality factors are as follows. Openness measures open-mindedness to new experiences, including traits such as creativity, insightfulness, and originality. Conscientiousness measures efficiency and organization, including resourcefulness and intelligence. Extraversion measures sociability, including outgoingness, self-confidence, and aggression. Agreeableness measures friendliness and compassion, including trustworthiness, compliance, and modesty. Lastly, neuroticism measures emotional vulnerability, including moodiness, hostility, self-consciousness, and impulsivity [11].

In respect of individuals' personalities, the Big Five are quantified using the NEO-PI psychometric inventory [3]. A common methodology of music-personality studies associates NEO-PI results with music-preference tests (e.g. for genres). Results from existing studies have revealed many relationships between the Big Five and musical preferences, which will now be overviewed.

Individuals with high openness typically prefer genres such as blues and jazz, while avoiding pop and country [19]. They also enjoy a wider variety of musical genres overall [15]. High conscientiousness has been linked to soul and funk [19]. Extraverts prefer pop and rap [19], which commonly occur in social situations, and thus may appeal to those high in extraversion [14, 15]. High agreeableness is associated with soundtracks (e.g. of films). The fifth factor, neuroticism, predicts preference for genres with exaggerated bass, such as dance [10].

The current study examined music and personality in terms of music-consumption patterns. The primary pattern we studied was genre exclusivity—a measure of the variety of genres in users' music collections. Genre exclusivity can be thought of as a scale with two extremes. The lower end contains homogenous music collections with very few genres (referred to as "genre exclusive"); the upper end contains heterogeneous music collections with many well-represented, distinct genres (referred to as "genre inclusive"). We investigated the link between genre homogeneity/heterogeneity, musical preference and factors within the Big Five, and in so doing evaluated the predictive power of personality on genre exclusivity.

1.1 MixRadio Database

This study utilized a music-download database, the majority of which were made onto Nokia mobile phones. The data became accessible through a data sharing agreement between McMaster University and the Nokia Corporation



© Jotthi Bansal, Matthew Woolhouse.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Jotthi Bansal, Matthew Woolhouse. "Predictive power of personality on music-genre exclusivity", 16th International Society for Music Information Retrieval Conference, 2015.

which begun in 2012. In January 2015 the Nokia division responsible for music became a separate entity under the name MixRadio. Henceforth, we referred to the data as coming from the MixRadio database.

The MixRadio database contains downloads from 33 countries across the globe¹ and spans from 2007 to September of 2014. Currently, the database contains the metadata of 1.36 billion individual downloads from over 17 million MixRadio users.² MixRadio users had free access to unlimited amounts of music on online music stores, meaning they could explore musical genres without cost constraints. Each download’s metadata includes information such as track name, artist, album, genre, user ID (anonymous), date, (local) time and country. Open source databases including MusicBrainz (the open music encyclopedia) [9] and The Echo Nest [6] are used to supplement download metadata and enrich the database. Examples of supplemented information from additional databases include track-release date, tempo, key, mode, time signature and instrumentation. The data are arranged into a relational database management system and queried using the open-source MySQL implementation of SQL [18], and the Python Database API [9], enabling more extensive, iterative analyses to be undertaken.

Our first study used the MixRadio database to explore music consumption behaviours of genre-defined subgroups of users. We referred to these subgroups as “x-heads”, where “x” was a user’s most downloaded genre. As genre is the most commonly used musical classifier [16], we assumed genre to be a reliable marker of musical interest.

The second study examined the relationship between genre exclusivity of x-head subgroups and Big Five personality factors. We correlated measures of genre exclusivity with measures from an existing study associating the Big Five with preference for particular genres. We hypothesized that openness values would positively correlate with genre inclusivity (having a heterogeneous music collection). In other words, those high in openness should also be open to numerous genres. Previous literature has found that those high in openness tend to prefer diverse musical genres [15]. We conjectured that the remaining Big Five factors—extraversion, neuroticism, agreeableness and conscientiousness—would not correlate with genre exclusivity, due to lack of evidence of this in previous studies.

1.2 Study Parameters

As existing music-personality study focused on Western populations, we elected only to include user data from European countries (14 countries in total): Austria, Finland, France, Germany, Great Britain, Ireland, Italy, Netherlands, Norway, Poland, Portugal, Spain, Sweden and Switzerland. Downloads were also limited to the ten most com-

monly used genres in existing music and personality studies: classical, country, dance, folk, indie, jazz, metal, pop, rap and rock. Finally, to ensure robust measures of genre exclusivity, only users with between 10 and 5,000 downloads were included; heuristically, we decided that fewer than ten would be an insufficient sample size; greater than five-thousand might indicate that a user was simply a musical “stamp collector”.

2. STUDY 1.1

We used the MixRadio database to explore genre exclusivity in genre-defined subgroups of users. Each user in the study was categorized as an “x-head”, where x was the most popular genre within a user’s download collection. For example, if a user’s total collection contained 40 metal downloads, and 10 dance, they were defined as a “metal-head”, and placed within the metal-head subgroup. If no genre was more popular than any other in a user’s collection (e.g. 10 pop and 10 rock), the user was classified based on whichever genre they downloaded first. The raw counts per genre were obtained for each user, and a (normalized) level of genre exclusivity per user calculated by dividing the SD of the genre counts by their total number of downloads.

So as to weigh each country’s contribution to genre exclusivity equally, users in x-head subgroups were then subdivided based on user-country, and a median SD per x-head subgroup per country was calculated; this value was called “x-med”. For each x-head subgroup the x-med was derived from fourteen SD values (one per country). X-head subgroups were ranked based on their degree of genre exclusivity, i.e. x-med value. The lower the x-med, the more genre inclusive the x-head subgroup; the higher the x-med value, the more genre exclusive the x-head subgroup.

2.1 Results

Table 1 displays x-med values for x-head subgroups from most inclusive on left, to most exclusive on right. Indie-heads, who had the lowest x-med (0.137), were the most genre inclusive subgroup, while pop-heads who had the highest x-med (0.200) were the most genre exclusive. A more detailed look at x-head subgroups’ collections based on genre is discussed in Study 1.2 below.

3. STUDY 1.2

This study examined how x-head subgroups consumed music from individual genres. Specifically, we looked at pairs of x-head subgroups and examined the degree to which both x-head subgroups consumed each other’s main, group-defining genre. Equation (1) calculates the degree to which x-head subgroups consumed each genre.

$$S_{j,i} = \frac{1}{N} \sum_{j=1}^N \left(\frac{C_{i,j}}{C_{j,j} + C_{i,j}} \right) \quad (1)$$

¹ Argentina, Australia, Austria, Brazil, Britain, Canada, Chile, China, Finland, France, Germany, India, Indonesia, Ireland, Italy, Malaysia, Mexico, Netherlands, Norway, Poland, Portugal, Russia, Saudi Arabia, Singapore, South Africa, Spain, Sweden, Switzerland, Thailand, Turkey, United Arab Emirates, United States of America, Venezuela

² This represents only a portion of MixRadio’s total database, and is not indicative of market share.

X-HEAD	Indie	Jazz	Folk	Country	Classical	Rock	Metal	Rap	Dance	Pop
X-MED	0.137	0.142	0.158	0.160	0.161	0.165	0.167	0.178	0.181	0.200

Table 1. Percentage of genres in each x-head subgroup's collection compared to their main genre.

X-HEAD	GENRE										X-HEAD MEDIAN
	Rock	Pop	Indie	Metal	Rap	Dance	Folk	Jazz	Classical	Country	
Rock		29.1	13.1	15.5	6.1	6.2	2.6	2.7	2.5	2.2	6.1
Pop	20.9		6.1	6.0	8.1	9.9	2.0	1.9	2.6	1.6	6.0
Indie	29.4	25.8		5.7	7.9	11.5	2.3	1.8	1.7	1.3	5.7
Metal	26.8	19.5	5.1		5.8	4.9	1.1	0.7	0.9	0.8	4.9
Rap	11.7	23.4	4.5	5.7		11.2	0.6	0.0	0.5	0.5	4.5
Dance	12.7	27.5	7.6	3.9	11.7		0.5	0.8	1.0	0.5	3.9
Folk	16.8	22.9	4.4	4.5	3.4	3.0		1.8	3.0	2.1	3.4
Jazz	17.3	22.1	3.8	2.9	2.5	2.8	2.7		7.7	2.8	2.9
Classical	11.9	21.8	3.4	1.7	1.6	2.5	2.7	6.1		1.1	2.7
Country	16.2	21.7	3.4	3.5	2.1	2.1	2.3	1.4	1.5		2.3
GENRE MEDIAN	16.8	22.9	4.5	4.5	5.8	4.9	2.3	1.8	1.7	1.3	

Figure 1. Percentage of genres in each x-head subgroup's collection compared to their main genre.

$C_{i,j}$ = count of genre i in x-head j 's collection

N = number of x-heads

$S_{j,i}$ = the value of n^{th} row and i^{th} column (in particular,

$S_{j,i}$ is a measure of the average relative proportion of genre i in x-head j 's collection)

Each value of $S_{j,i}$ refers to a cell shown in Figure 1.

3.1 Results

Figure 1 shows the degrees to which x-head subgroups consumed other genres. The left-axis lists x-head subgroups; the top-axis lists the genres they consumed. The darker the cell, the greater the degree of genre consumption. The x-head medians listed in the far right column are the median percentages of the genres consumed by x-head subgroups. The genre medians listed along the bottom are the median percentages that each genre is consumed by the x-head subgroups. Figure 1 is symmetrical along its diagonal axis (diagonal line of white cells). By comparing each side of the diagonal axis, relationships between genre pairs can be explored. For example, rock-heads and pop-heads consumed the greatest percentage of each another's genres: rock-heads consumed 29.1% of pop, pop-heads consumed 20.9% of rock.

Various "classes" of relationships appeared based on the degree of genre consumption by pairs of x-head subgroups. Some x-head subgroup pairs consumed equal amounts of each other's main genre, and therefore had symmetrical relationships (same-shaded cells across the diagonal axis, e.g. rap and metal). Some x-head subgroup pairs consumed unequal amounts of each other's main genre, and

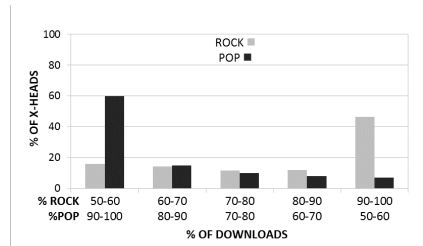


Figure 2. H-H consumption relationship between pop-heads and rock-heads.

therefore had asymmetrical relationships (differently shaded cells across the diagonal axis, e.g. indie and pop).

Symmetrical relationships were also classified as "hot" or "cold" based on the volume of consumption between two x-head subgroups. Symmetrically hot relationships occurred when both x-head subgroups downloaded significant amounts of each other's main genre. Symmetrically cold relationships occurred when neither x-head subgroup downloaded significant amounts of each other's main genre. Overall, three categories of x-head relationships were identified and are defined below using example pairs of x-head subgroups.

3.1.1 Symmetrical hot relationships (H-H)

Pairs of x-head subgroups downloading significant and approximately equally amounts of one another's main genre, e.g. rock-heads and pop-heads (Figure 2).

Figure 2 shows the composition of rock-heads' (grey) and pop-heads' (black) collections when comparing only the proportion of rock and pop downloads they each consumed. The x-axis displays a series of bins which describe the proportion of rock and pop downloads in x-heads' collections (totalling 100%). The y-axis is the percentage of x-heads that fit into the specifications of each bin on the x-axis. There are two sets of horizontal-axis labels: the upper labels (% Rock) show the proportion of rock downloads represented in rock-heads' collections. The remaining proportion consists of pop downloads. For example, the grey column in the % Rock bin marked "50-60" shows the percentage of rock-heads whose collection contained approximately 50-60% rock downloads and 40-50% pop downloads. The lower labels (% Pop) show the proportion of pop downloads represented in pop-heads' collections. The remaining percentage consists of rock downloads.

H-H relationships are represented in Figure 1 by diagonally related dark-shaded squares. X-head subgroup pairs with H-H relationships can thought of as being mutually

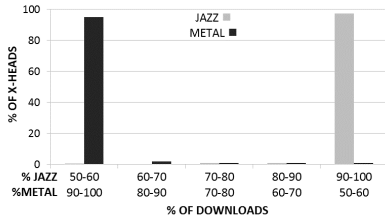


Figure 3. C-C consumption relationship between jazz-heads and metal-heads.

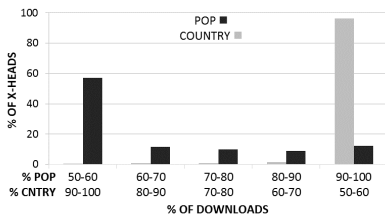


Figure 4. H-C consumption relationship between pop-heads and country heads.

inclusive, and vice versa for light-shaded squares.

3.1.2 Symmetrical cold relationships (C-C)

Pairs of x-head subgroups who downloaded roughly equal, but insignificant amounts of the each others’ main genre, e.g. jazz-heads and metal-heads (Figure 3).

The axes in Figure 3 are the same as those in Figure 2, but represent jazz-heads and metal-heads instead. Bar heights in Figure 3 reveal that a majority jazz-heads and metal-heads had a ratio of 90-100% of their main genre and 0-10% of the other. Very few jazz-heads or metal-heads downloaded equal amounts of both genres. C-C relationships are represented in Figure 1 by diagonally related light-shaded squares. X-head subgroup pairs with C-C relationships can be thought of as being mutually exclusive.

3.1.3 Asymmetrical hot-cold relationships (H-C)

Pairs of x-head subgroups who consumed each other’s main genre unequally, e.g. pop-heads and country-heads (Figure 4).

The axes in Figure 4 are the same as those in Figures 2 and 3, but represent pop-heads and country-heads. Bar heights in Figure 4 revealed that many country-heads consumed large amounts of both pop and country music. However, a majority of pop-heads did not consume significant amounts of country music. H-C relationships are represented in Figure 1 by diagonally related cells, between

which there is a mismatch in shading, i.e. light grey to dark grey.

3.2 Study 1 Conclusions

In Study 1.1, x-head subgroups ranked from genre exclusive to inclusive in the following order: pop, dance, rap, metal, rock, classical, country, folk, jazz, and indie. Intriguingly, this ranking is consistent with previous literature indicating that individuals who prefer jazz and folk music rank highly in the Big Five factor of openness, which has been linked to genre inclusivity. Those who are high in openness also tend to avoid genres like pop; pop-heads were found to be the most genre exclusive. Therefore, study 1.1 results preliminarily hinted at links between genre exclusivity and aspects of personality.

In Study 1.2, pairs of x-head subgroups were compared based on their consumption of one another’s main genre. Some x-head subgroup pairs were mutually inclusive of one another (H-H), while others were mutually exclusive (C-C). Remaining x-head pairs consumed each other’s main genres unequally (H-C).

4. STUDY 2

Study 2 examined links between genre exclusivity and the Big Five personality factors. Our measures of genre exclusivity (median SD per x-head subgroup per country) were correlated with measures of Big Five personality factors that had previously been associated with certain genres from Zweigenhaft (2008) [19].

Zweigenhaft had subjects complete the NEO-PI and a version of the STOMP (Short Test of Music Preferences), [16]. Measures of Big Five personality and music preference were then correlated. We used the correlation values between Big Five factors and genres from Zweigenhaft (2008), and correlated them with levels genre exclusivity from Study 1.1 (14 country values per x-head subgroup).

4.1 Results

A significant, negative correlation existed between genre exclusivity and genres associated with openness (Figure 5: n = 140; r = -0.37; two-tailed, p < 0.001) and agreeableness (Figure 6: n = 140; r = -0.32; two-tailed, p < 0.001). That is, genre-openness associations and genre-agreeableness associations in Zweigenhaft (2008) predicted genre inclusivity in x-head subgroups. There were no significant correlations between extraversion, conscientiousness and neuroticism with genre exclusivity.

Figures 5 and 6 show relationships between openness and agreeableness with genre exclusivity. The horizontal-axes display degree of genre exclusivity for x-head subgroups (median SD of x-heads’ music collections based on genre). Each x-head subgroup (listed down the right legend) is represented with a different shade of grey. Horizontally positioned markers with the same shade are the median SDs per x-head subgroup for each of the 14 countries included in the study. The height of the markers cor-

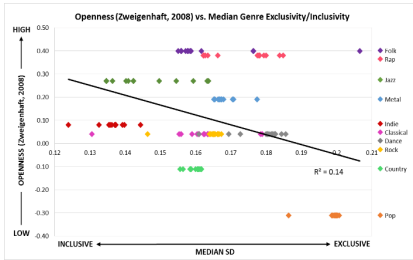


Figure 5. X-head genre exclusivity against genre-openness associations in Zweigenhaft (2008).

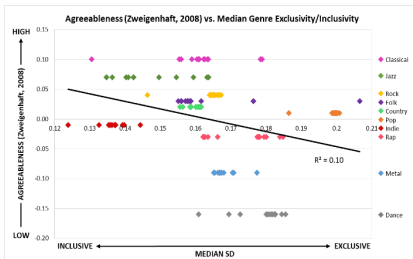


Figure 6. X-head genre exclusivity against genre-agreeableness associations in Zweigenhaft (2008).

responds to the degree of openness and agreeableness for each genre in Zweigenhaft (2008), shown on the y-axes.

4.2 Study 2 Conclusions

Genre-openness and -agreeableness associations from Zweigenhaft (2008) predicted genre inclusivity in x-head subgroups; if you score high in openness and/or agreeableness you are likely to have more genres within your music collection. Conscientiousness, extraversion and neuroticism are not predictors of genre exclusivity.

5. DISCUSSION

Study 1 explored overall genre exclusivity of x-head subgroups. Study 1.2 revealed the pairwise relationships between x-head subgroups. Some of these relationships were one-sided; only one of the two x-head subgroups consumed music from the other's main genre. While others were more equitable; both x-head subgroups consumed each other's main genre equally.

Study 2 revealed links between genre exclusivity and personality; openness and agreeableness predicted preference for a wide range of genres. Breaking down openness and agreeableness based on their traits reveals possible reasons for their relationship with genre exclusivity. Openness is a general willingness to encounter new experi-

ences, and different musical styles certainly constitute new experiences. If someone is open to new experiences, they also seem to be open to new musical genres. Those high in openness tend to break from the rules of social boundaries [5] and may not fear venturing outside of Western-cultured musical norms. Those high in openness often dislike ubiquitous genres like pop [19], tending, instead, to explore less commercial musical styles. Moreover, they use music for cognitive and rational purposes, such as intellectual stimulation, and focus more on the quality, complexity and performance [1]. Exploration of numerous genres may satisfy their desire for these musical properties.

The ability of agreeableness to predict genre exclusivity was unanticipated—few studies have found this factor to be a reliable predictor of musical preference. However, agreeableness encompasses traits such as compliance [19], so perhaps those who are agreeable may also be “compliant” to various musical genres. To test this theory, associations between traits of agreeableness and genre exclusivity would have to be examined.

5.1 Limitations

Given that our data were derived predominantly from mobile-phone users, it may be problematic to generalize our findings to those who acquire music from other sources. Moreover, Studies 1 and 2 were restricted to European countries, again, limiting result generalizability. Since personality [17] and musical preferences [16] vary between countries, our results may not be globally consistent.

A second population-based limitation relates to socioeconomic variance between individuals and countries. The users in the MixRadio database are biased to those who can afford a Nokia mobile phone. Despite this, Nokia has historically made a range of models to appeal to different market sectors. Therefore, although the self-selected users in our study may not be fully representative, it is assumed that they are relatively widely distributed throughout the populations of the countries within our study.

A third limitation arises when associating genre-personality correlations from Zweigenhaft (2008) with measures of genre exclusivity: the subject group tested in Zweigenhaft (2008) are not the same as the MixRadio user population. However, without gathering personality information directly from MixRadio users, genre-personality correlations were the most suitable measure to associate with genre exclusivity.

Additionally, given that pop is the commonest genre, it is perhaps not surprising that most pairwise relationships with pop are asymmetrical and that pop is the most popular genre for non-pop-heads. However, despite this limitation the method adopted (as shown in Figure 1) does at least indicate instances where x-head subgroups consume different amounts of another genre relative to one another. For example, relatively speaking, pop-heads consume less country than jazz-heads.

A possible methodological complication relates to the way in which x-heads are defined based upon most downloaded genre. That is, we assume that users' genre distri-

butions represent genuine musical preferences, which, although likely to be the case, is not known for certain. In other words, our notion of genre popularity could be a misrepresentation of musical tastes.

5.2 Implications

Information about x-head genre exclusivity is a valuable resource in music marketing and recommender systems. For example, a MixRadio user purchased a large quantity of country songs. For example, based on results from Study 1.2, country-heads would appear to be susceptible to pop, although, given the asymmetrical relationship between these genres, the reverse seems not to be the case (country-heads consume pop, but pop-heads do not consume country). Understanding each side of x-head relationships could be useful in avoiding misguided recommendations.

Moreover, understanding the link between personality and genre consumption may prove useful in music marketing. If a user were to complete a Big Five personality questionnaire upon signing up with a music service, information concerning openness and agreeableness could be factored into recommendations; e.g. wide range of obscure genres for those open and/or agreeable, and vice versa.

5.3 Future Studies

The reasons underpinning genre inclusivity or exclusivity can be examined further. For example, perhaps certain genres are downloaded in tandem due to similar acoustic properties such as tempo, key, instrumentation, or metrical structure. Feature analysis and genre preference will be a target of future studies.

Our new-found links between the Big Five and genre exclusivity mark the beginning of explorations on personality and music consumption. Other types of exclusivity relationships may also be linked to personality traits, including artist exclusivity (the number of artists in a user's collection), tempo exclusivity (variety of tempos in a user's collection), or release-date exclusivity (the era from which a musical collection stems). We hope to examine these factors, other factors, and their possible links to personality.

6. CONCLUSION

By analyzing a subset of mobile phone music-download data, the current study revealed information concerning musical-genre consumption. Genre-defined subgroups of users acquired music in unique and distinctive ways, with varying degrees of acceptance for other musical styles. Overall, genre exclusivity was most consistently associated with the Big Five personality factor of openness, which supports similar research in existing music-personality studies. Genre exclusivity was also linked to agreeableness, adding a new finding to the music-personality literature. Overall, the more open or agreeable you are, the more genre inclusive, or heterogeneous, your musical tastes.

The current study introduced a novel big-data methodology to music-personality studies, which we will continue

to utilize. With access to ever-growing music-download databases, the predictive power of personality on genre exclusivity is an exciting and expanding field of music-consumption research.

7. REFERENCES

- [1] T. Chamorro-Premuzic, and A. Furnham. Personality and music: Can traits explain how people use music in everyday life? *British Journal of Psychology*, 98(2): 175–85, 2007.
- [2] P. T. Jr. Costa, and R. R. McCrae. *The NEO Personality Inventory Manual*, Psychological Assessment Resources, Odessa, 1985.
- [3] P. T. Jr. Costa, and R. R. McCrae. Four ways the five factors are basic. *Personality and Individual Differences*, 13(6): 653–665, 1992.
- [4] S. J. Dollinger. Research note: Personality and music preference: Extraversion and excitement seeking or openness to experience? *Psychology of Music*, 21(1): 73–77, 1993.
- [5] S. J. Dollinger, L. A. Orf, and A. E. Robinson. Personality and campus controversies: Preferred boundaries as a function of openness to experience. *The Journal of Personality*, 125(4): 399–406, 1991.
- [6] T. Jehan, and B. Whitman. The Echo Nest. [Data set]. Retrieved from <http://the.echonest.com/>, 2005.
- [7] O. P. John and S. Srivastava. Big five trait taxonomy: History, measurement, and theoretical perspectives. In L. Pervin & O. P. John (Eds.), *Handbook of personality: Theory and research (2nd ed.)* (pp. 102–138), Guilford, Berkeley, 1999.
- [8] R. Kaye. MusicBrainz Database [Data set]. Retrieved from [http://musicbrainz.org/doc/MusicBrainz_Data base](http://musicbrainz.org/doc/MusicBrainz_Data_base), 2000.
- [9] M. A. Lemberg. Python database API specification v2.0. *Python Enhancement Proposal*, 249, 2008.
- [10] W. McCown, R. Keiser, S. Mulhearn, and D. Williamson. The role of personality and gender in preference for exaggerated bass in music. *Personality and Individual Differences*, 23(4): 543–547, 1997.
- [11] R. R. McCrae, and P. T. Costa, P. T. Validation of the five-factor model of personality across instruments and observers. *Journal of Personality and Social Psychology*, 52(1): 81–90, 1987.
- [12] A. C. North, and D. J. Hargreaves. Music and adolescent identity. *Music Education Research*, 1(1): 75–92, 1999.
- [13] E. Payne. Musical taste and personality. *The British Journal of Psychology*, 58(1): 133–138, 1967.

- [14] J. L. Pearson, and S. J. Dollinger. Music preference correlates of Jungian types. *Personality and Individual Differences*, 36(5): 1005–1008, 2004.
- [15] D. Rawlings, and V. Ciancarelli. Music preference and the five-factor model of the NEO personality inventory. *Psychology of Music*, 25(2): 120–132, 1997.
- [16] P. J. Rentfrow, and S. D. Gosling. The do re mi?s of everyday life: The structure and personality correlates of music preferences. *Journal of Personality and Social Psychology*, 84(6): 1236–1256, 2003.
- [17] D. P. Schmitt, J. Allik, R. R. McCrae, and V. Benet-Martinez. The geographic distribution of big five personality traits: Patterns and profiles of human self-description across 56 nations. *Journal of Cross-Cultural Psychology*, 38(2): 173–212, 2007.
- [18] P. Weinberg, J. Groff, A. Opper, and A. Davenport. *SQL, the Complete Reference*. McGraw-Hill, New York, 2010.
- [19] R. L. Zweigenhaft. A do re mi encore. *Journal of Individual Differences*, 29(1), 45–55, 2008.

A COMPARISON OF SYMBOLIC SIMILARITY MEASURES FOR FINDING OCCURRENCES OF MELODIC SEGMENTS

Berit Janssen
Meertens Institute,
Amsterdam
berit.janssen
@meertens.knaw.nl

Peter van Kranenburg
Meertens Institute,
Amsterdam
peter.van.kranenburg
@meertens.knaw.nl

Anja Volk
Utrecht University,
the Netherlands
a.volk@uu.nl

ABSTRACT

To find occurrences of melodic segments, such as themes, phrases and motifs, in musical works, a well-performing similarity measure is needed to support human analysis of large music corpora. We evaluate the performance of a range of melodic similarity measures to find occurrences of phrases in folk song melodies. We compare the similarity measures correlation distance, city-block distance, Euclidean distance and alignment, proposed for melody comparison in computational ethnomusicology; furthermore Implication-Realization structure alignment and B-spline alignment, forming successful approaches in symbolic melodic similarity; moreover, wavelet transform and the geometric approach Structure Induction, having performed well in musical pattern discovery. We evaluate the success of the different similarity measures through observing retrieval success in relation to human annotations. Our results show that local alignment and SIAM perform on an almost equal level to human annotators.

1. INTRODUCTION

In many music analysis tasks, it is important to query a large database of music pieces for the occurrence of a specific melodic segment: which pieces by Rachmaninov quote *Dies Irae*? Which bebop jazz improvisers used a specific Charlie Parker lick in their solos? How many folk song singers perform a melodic phrase in a specific way?

In the present article, we compare a range of existing similarity measures with the goal of finding occurrences of melodic segments in a corpus of folk song melodies. This is a novel research question, evaluated on annotations which have been made specifically for this purpose. The insights gained from our research on the folk song genre can inform future research on occurrences in other genres.

We evaluate similarity measures on a set of folk songs, in which human experts annotated phrase similarity. We

use these annotations as evidence for occurrences of melodic segments in related songs. If we know that a similarity measure is successful in finding the annotated occurrences in this set, we infer that the measures will be successful for finding correct occurrences of melodic segments of phrase length in a larger dataset of folk songs as well. We describe the dataset in more detail in Section 2.


In computational ethnomusicology various methods for comparing folk song melodies have been suggested: as such, correlation distance [12], city-block distance and Euclidean distance [14] have been considered promising. Research on melodic similarity in folk songs also showed that alignment measures reproduce human judgements on agreement between melodies well [16].

As this paper focusses on similarity of melodic segments rather than whole melodies, recent research in musical pattern discovery is also of particular interest. Two well-performing measures in the associated MIREX challenge of 2014 [7, 17] have shown success when evaluated on the Johannes Kepler University segments Test Database (JKUPDT).¹ We test whether the underlying similarity measures of the pattern discovery methods also perform well in finding occurrences of melodic segments.

Additionally, we apply the most successful similarity measures from the MIREX symbolic melodic similarity track in our research. The best measure of MIREX 2005 (Grachten et al. [4]), was evaluated on RISM incipits, which are short melodies or melodic segments, therefore relevant for our task. In recent MIREX editions the algorithm by Urbano et al. [15] has been shown to perform well on the EsAC folk song collection.²

We present an overview of the compared similarity measures in Table 1, listing the music representations to which these measures have been originally applied, and which we therefore also use in our comparisons. Moreover, we include information on the research fields from which the measures are taken, the database on which they were evaluated, if applicable, and a bibliographical reference to a relevant paper. We describe the measures in Section 3.

We evaluate the different measures by comparison with human annotations of phrase occurrence, through quanti-

 © Berit Janssen, Peter van Kranenburg, Anja Volk.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Berit Janssen, Peter van Kranenburg, Anja Volk. "A comparison of symbolic similarity measures for finding occurrences of melodic segments", 16th International Society for Music Information Retrieval Conference, 2015.

¹ http://www.music-ir.org/mirex/wiki/2014:Discovery_of_Repeated_Themes_%26_Sections_Results

² <http://www.esac-data.org>

Similarity measure	Music representations	Research field	Dataset	Authors
Correlation distance (CD)	duration weighted pitch sequence	Ethnomusicology	-	[12]
City block distance (CBD)	pitch sequence	Ethnomusicology	-	[14]
Euclidean distance (ED)	pitch sequence	Ethnomusicology	-	[14]
Local alignment (LA)	pitch sequence	Ethnomusicology	MTC	[16]
Structure induction (SIAM)	pitch / onset	MIR	JKUPTD	[7]
Wavelet transform (WT)	duration weighted pitch sequence	MIR	JKUPTD	[17]
B-spline alignment (BSA)	pitch sequence	MIR	EsAC	[15]
I-R structure alignment (IRSA)	pitch, duration, metric weight	MIR	RISM	[4]

Table 1. An overview of the measures for music similarity compared in this research, with information on the authors and year of the related publication, and which musical data the measures were tested on, if applicable.

fyng the retrieval measures precision, recall and F1-score, and the area under the receiver-operating characteristic curve. The evaluation procedure is described in detail in Section 4.

The remainder of this paper is organised as follows: first, we describe our corpus of folk songs and the annotation procedure. Next, we give details on the compared similarity measures, and the methods used to implement the similarity measures. We describe our evaluation procedure before presenting the results, finally discussing the implications of our findings and concluding steps for future work.

2. MATERIAL

We evaluate the similarity measures on a corpus of Dutch folk songs, MTC-ANN 2.0, which is part of the Meertens Tune Collections [5]. MTC-ANN 2.0 contains 360 orally transmitted melodies, which have been transcribed from recordings and digitized in various formats. Various meta-data have been added by domain experts, such as the tune family membership of a given melody: the melodies were categorized into groups of variants, or tune families. The variants belonging to a tune family are considered as being descended from the same ancestor melody [1]. We parse the `**kern` files as provided by MTC-ANN 2.0 and transform the melodies and segments into the required music representations using `music21` [2].

Even though MTC-ANN 2.0 comprises very well documented data, there are some difficulties to overcome when comparing the digitized melodies computationally. Most importantly, the transcription choices between variants can be different: where one melody is notated in 3/4, and with a melodic range from D4 to G4, another transcriber may have chosen a 6/8 meter, and a melodic range from D3 to G3. This means that notes which are perceptually very similar might be hard to match based on the digitized information. Musical similarity measures might be sensitive to these differences, or they might be transposition or time dilation invariant, i.e. work equally well under different pitch transpositions or meters.

Of these 360 melodies categorized into 26 tune families, we asked three Dutch folk song experts to annotate similarity relationships between phrases within tune families. The

annotators judged the similarity of phrases of 213 melodies belonging to 16 tune families, amounting to 1084 phrase annotations in total. The phrases contain, on average, nine notes, with a standard deviation of two notes. The dataset with its numerous annotations is publicly available.³

For each tune family, the annotators compared all the phrases within the tune family with each other, and gave each phrase a label consisting of a letter and a number. If two phrases were considered “almost identical”, they received exactly the same label; if they were considered “related but varied”, they received the same letter, but different numbers; and if two phrases were considered “different”, they received different letters. See an annotation example in Figure 1.

The three domain experts worked independently on the same data. To investigate the subjectivity of similarity judgements, we measured the agreement between the three annotators’ similarity judgements using Fleiss’ Kappa, which yielded $\kappa = 0.73$, constituting substantial agreement.

The annotation was organized in this way to guarantee that the task was feasible: judging the occurrences of hundreds of phrases in dozens of melodies (14714 comparisons) would have been much more time consuming than assigning labels to the 1084 phrases, based on their similarity. Moreover, the three levels of annotation facilitate evaluation for two goals: finding only almost identical occurrences, and finding also varied occurrences. These two goals might require quite different approaches.

We focus on finding almost identical occurrences: if for a given query phrase q in one melody, at least one phrase r with exactly the same label (letter and number) appears in another melody s of the same tune family, we consider it an occurrence of melodic segment q in s . Conversely, if there is no phrase with exactly the same label as q in melody s , this constitutes a non-occurrence.

For all phrases and all melodies, within their respective tune families, we observe whether the annotators agree on occurrence or non-occurrence of query phrases q in melodies s . The agreement for these judgements, 14714 in total, was analyzed with Fleiss’ Kappa, with the result $\kappa = 0.51$ denoting moderate agreement. This highlights the ambig-

³ <http://www.liederenbank.nl/mtc/>

Figure 1. An example for two melodies from the same tune family with annotations.

Annotators	Precision	Recall	F1-score
1 and 2	0.745	0.763	0.754
1 and 3	0.803	0.75	0.776
2 and 3	0.788	0.719	0.752

Table 2. The retrieval scores between annotators. For instance, annotator 2 agrees to 75% with the occurrences detected by annotator 1. The scores are symmetric.

ity involved in finding occurrences of melodic segments.

To compare the annotators’ agreement with the performance of the similarity measures in the most meaningful way, we also compute the precision, recall and F1-score of each annotator in reproducing the occurrences detected by another annotator. Table 2 gives an overview of these retrieval scores. A higher retrieval score for a given similarity measure would indicate overfitting to the judgements of one individual annotator.

3. COMPARED SIMILARITY MEASURES

In this section, we present the eight compared similarity measures. We describe the measures in three subgroups: first, measures comparing fixed-length note sequences; second, measures comparing variable-length note sequences; third, measures comparing more abstract representations of the melody.

For our corpus, as melodies are of similar length, we can transpose all melodies to the same key using pitch histogram intersection. For each melody, a pitch histogram is computed with MIDI note numbers as bins, with the count of each note number weighed by its total duration in a melody. The pitch histogram intersection of two histograms h_q and h_r , with shift σ is defined as

$$PHI(h_q, h_r, \sigma) = \sum_{k=1}^l \min(h_{q,k+\sigma}, h_{r,k}), \quad (1)$$

where k denotes the index of the bin, and l the total number of bins. We define a non-existing bin to have value zero. For each tune family, we randomly pick one melody and for each other melody in the tune family we compute the σ that yields a maximum value for the histogram intersection, and transpose that melody by σ semitones.

Some similarity measures use note duration to increase precision of the comparisons, others discard the note du-

ration, which is an easy way of dealing with time dilation differences. Therefore, we distinguish between music representation as *pitch sequences*, which discard the durations of notes, and *duration weighted pitch sequences*, which repeat a given pitch depending on the length of the notes. We represent a quarter note by 16 pitch values, an eighth note by 8 pitch values, and so on. Onsets of small duration units, especially triplets, may fall between these sampling points, which shifts their onset slightly in the representation. Besides, a few similarity measures require music representation as *onset*, *pitch pairs*, or additional information on metric weight.

3.1 Similarity Measures Comparing Fixed-Length Note Sequences

To formalize the following three measures, we refer to two melodic segments q and r of length n , which have elements q_i and r_i . The measures described in this section are distance measures, such that lower values of $dist(q, r)$ indicate higher similarity. Finding an occurrence of a melodic segment within a melody with a fixed-length similarity measure is achieved through the comparison of the query segment against all possible segments of the same length in the melody. The candidate segment which is most similar to the query segment is retained as a match. The implementation of the fixed-length similarity measures in Python is available online.⁴ It uses the *spatial.distance* library of scipy [10].

Scherrer and Scherrer [12] suggest correlation distance to compare folk song melodies, represented as duration weighed pitch sequences. Correlation distance is independent of the transposition and melodic range of a melody, but in the current music representation, it is affected by time dilation differences.

$$dist(q, r) = 1 - \frac{\sum_{i=1}^n (q_i - \bar{q}) \cdot \sum_{i=1}^n (r_i - \bar{r})}{\sqrt{\sum_{i=1}^n (q_i - \bar{q})^2} \cdot \sqrt{\sum_{i=1}^n (r_i - \bar{r})^2}} \quad (2)$$

Steinbeck [14] proposes two similarity metrics for the classification of folk song melodies: city-block distance and Euclidean distance (p.251f.). He suggests to compare pitch sequences, next to various other features of melodies such as their range, or the number of notes in a melody. As we are interested in finding occurrences of segments

⁴ <https://github.com/BeritJanssen/MelodicOccurrences>

rather than comparing whole melodies, we analyze pitch sequences.

City-block distance and Euclidean distance are not transposition invariant, but as they are applied to pitch sequences, they are time dilation invariant. All the fixed-length measures in this section will be influenced by small variations affecting the number of notes in a melodic segment, such as ornamentation. Variable-length similarity measures, discussed in the following section, can deal with such variations more effectively.

3.2 Similarity Measures Comparing Variable-Length Note Sequences

To formalize the following three measures, we refer to a melodic segment q of length n and a melody s of length m , with elements q_i and s_j . The measures described in this section are similarity measures, such that lower values of $sim(q, s)$ indicate higher similarity. The implementation of these methods in Python is available online.⁴

Mongeau and Sankoff [8] suggest the use of alignment methods for measuring music similarity, and they have been proven to work well for folk songs [16]. We apply local alignment [13], which returns the similarity of a segment within a melody which matches the query best.

To compute the optimal local alignment, a matrix $A(i, j)$ is recursively filled according to equation 3. The matrix is initialized as $A(i, 0) = 0, i \in \{0, \dots, n\}$, and $A(0, j) = 0, j \in \{0, \dots, m\}$. $W_{insertion}$ and $W_{deletion}$ define the weights for inserting an element from melody s into segment q , and for deleting an element from segment q , respectively. $subs(q_i, s_j)$ is the substitution function, which gives a weight depending on the similarity of the notes q_i and s_j .

$$A(i, j) = \max \begin{cases} A(i-1, j-1) + subs(q_i, s_j) \\ A(i, j-1) + W_{insertion} \\ A(i-1, j) + W_{deletion} \\ 0 \end{cases} \quad (3)$$

We apply local alignment to pitch sequences. In this representation, local alignment is not transposition invariant, but it should be robust with respect to time dilation. For the insertion and deletion weights, we use $W_{insertion} = W_{deletion} = -0.5$, and we define the substitution score as

$$subs(q_i, s_j) = \begin{cases} 1 & \text{if } q_i = s_j \\ -1 & \text{otherwise} \end{cases} \quad (4)$$

The local alignment score is the maximum value in the alignment matrix, normalized by the number of notes n in the query segment.

$$sim(q, s) = \frac{1}{n} \max_{i,j} A(i, j) \quad (5)$$

Structure Induction Algorithms [7] formalize a melody as a set of points in a space defined by note onset and pitch, and perform well for musical pattern discovery [6]. They measure the difference between melodic segments

through so-called translation vectors. The translation vector \mathbf{T} between points in two melodic segments can be seen as the difference between the points q_i and s_j in onset, pitch space. As such, it is transposition invariant, but will be influenced by time dilation differences.

$$\mathbf{T} = \begin{pmatrix} q_{i,onset} \\ q_{i,pitch} \end{pmatrix} - \begin{pmatrix} s_{j,onset} \\ s_{j,pitch} \end{pmatrix} \quad (6)$$

The maximally translatable pattern (MTP) of a translation vector \mathbf{T} for two melodies q and s is then defined as the set of melody points q_i which can be transformed to melody points s_j with the translation vector \mathbf{T} .

$$MTP(q, s, \mathbf{T}) = \{q_i | q_i \in q \wedge q_i + \mathbf{T} \in s\} \quad (7)$$

We analyze the pattern matching method SIAM, defining the similarity of two melodies as the length of the longest maximally translatable pattern, normalized by the length n of the query melody:

$$sim(q, s) = \frac{1}{n} \max_{\mathbf{T}} |MTP(q, s, \mathbf{T})| \quad (8)$$

3.3 Similarity Measures Comparing Abstract Representations

The following three methods transform the melodic contour into a more abstract representation prior to comparison.

Velarde et al. [18] use wavelet coefficients to compare melodies: melodic segments are transformed with the Haar wavelet. The wavelet coefficients indicate whether there is a contour change at a given moment in the melody, and similarity between two melodies is computed through city-block distance of their wavelet coefficients. The method achieved considerable success for pattern discovery [17]. We use the authors' Matlab implementation to compute wavelet coefficients of duration weighed pitch sequences, and compute city-block distance between the coefficients of query segment and match candidates.

Through the choice of music representation and comparison of the wavelet coefficients, this is a fixed-length similarity measure sensitive to time dilation; however, it is transposition invariant.

Urbano et al. [15] transform note trigrams to a series of B-spline interpolations, which are curves fitted to the contours of the note trigrams. The resulting series of B-splines of two melodies are then compared through alignment. Different B-spline alignment approaches have performed well in various editions of MIREX for symbolic melodic similarity.⁵

We apply the ULMS2-ShapeL algorithm,⁶ using the most recent version, different from its original publication [15]. This algorithm discards the durations of the notes and returns the local alignment score of query segments and melodies. The score is normalized by the length

⁵ http://www.music-ir.org/mirex/wiki/2012:Symbolic_Melodic_Similarity_Results

⁶ <https://github.com/julian-urbano/MelodyShape>

n of the query segment. This similarity measure is of variable length, sensitive to time dilation, but transposition invariant.

Grachten’s method [4] relies on Implication-Realization (IR) structures, as introduced by Narmour [9] as basic units of melodic expectation. Grachten et al. transform melodies into IR structures using a specially developed parser. The similarity of melodies is then determined based on the alignment of the IR structures. This method was successful in the MIREX challenge for symbolic melodic similarity of 2005.⁷

In preparation of IR-structure alignment, we use Grachten’s [4] IR-parser, which takes the onset, pitch, duration and metric weight of a melody and infers the corresponding IR structures. To this end, we exclude all melodies which do not have annotated meter ($n = 65$), needed for the computation of metric weight, from the corpus. We align the IR-structures with the same insertion and deletion weights and the same substitution function as Grachten’s publication, but as we are interested in finding occurrences, we use local alignment rather than the original global alignment approach. Through the transformation of the note sequences to IR-structure sequences, this similarity measure is transposition invariant, but it is sensitive to time dilation and ornamentation, which might affect the detected IR-structures.

4. EVALUATION

We evaluate the potential success of a similarity measure through comparing the retrieved occurrences to the annotators’ judgements, separately for each annotator. Different thresholds on the similarity measures determine which matches are accepted as occurrences, or rejected as non-occurrences. For the distance measures (CD, CBD, ED, WT), matches with similarity values below the threshold, for the other measures, matches with similarity values above the threshold are considered occurrences.

The relationship between true positives and false positives for each measure is summarized in a receiver-operating characteristic (ROC) curve with the threshold as parameter. The area under the ROC curve (AUC) determines whether a similarity measure overall performs better than another, for which we calculate confidence intervals and statistical significance using DeLong’s method for paired ROC curves, based on U statistics [3, 11]. Furthermore, we report the maximally achievable retrieval measures precision, recall and F1-score with relation to the ground truth.

5. RESULTS

We have analyzed the results with respect to all annotators, resulting in the same ranking of the similarity measures. Due to space constraints, we report and discuss our results in relation to annotator 1. We show the ROC curves of the eight different measures in Figure 2, which display the true positive rate against the false positive rate at different

Measure	F1-score	Precision	Recall	AUC
Baseline	0.68	0.52	1	n/a
CD	0.68	0.51	1	0.549
CBD	0.68	0.51	1	0.574
ED	0.68	0.51	1	0.568
LA	0.73	0.7	0.78	0.790
SIAM	0.73	0.75	0.71	0.787
WT	0.69	0.57	0.87	0.732
BSA	0.72	0.65	0.81	0.776
IRSA	0.69	0.54	0.95	0.683

Table 3. Results of the compared similarity measures for different music representations: the maximal F1-score, the associated precision and recall, and the area under the ROC curve (AUC).

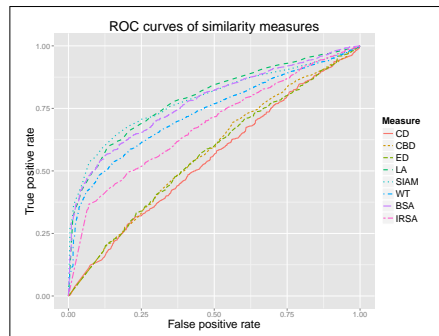


Figure 2. The ROC curves for the various similarity measures, showing the increase of false positive rate against the increase of the true positive rate, as a parameter of the threshold.

thresholds. The more of the higher left area a ROC curve covers in a graph, the better; this indicates that the two classes are better separable.

From Figure 2 it can be seen that the similarity measures suggested in computational ethnomusicology (CD, CBD, ED) perform only marginally above chance. IR-structure alignment and wavelet transform obtain better results, and B-spline alignment, local alignment and SIAM perform best.

We summarize the area under the ROC curve (AUC), the maximally achieved F1-score, as well as the associated precision and recall in Table 3. We include a baseline in this table which assumes that every compared melody contains an occurrence of the query segment, which leads to perfect recall, but poor precision, as the chance for a segment to occur in a given melody are only about 50%.

We compare the AUC values of the different measures in Figure 3, showing confidence intervals and significance of the pairwise differences between adjoining measures, indicated by stars ($*p < .5$, $**p < .01$, $***p < .001$).

⁷ http://www.music-ir.org/mirex/wiki/2005:Symbolic_Melodic_Similarity_Results

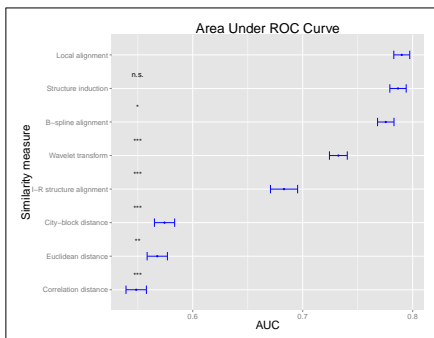


Figure 3. The area under the ROC curve of all similarity measures, ordered by the most successful to the least successful methods. The error bars indicate the confidence intervals, and significant difference between adjoining measures is indicated by stars (* $p < .5$, ** $p < .01$, *** $p < .001$).

6. DISCUSSION

Our results indicate that the distance measures (CD, CBD, ED) do not work very well, which contradicts the intuitions of the computational ethnomusicologists who propose them. This suggests that variations on pitch height and contour, which mostly affect these measures, are not the most informative aspect for human judgements on musical similarity. Embellishments of a note sequence through extra notes, for instance to accommodate slightly varied lyrics, on the other hand, would cause considerable decrease of measured similarity, while they will be perceived as minor variation, if at all by human listeners.

Measures from symbolic melodic similarity (BSA, IRSA) and pattern discovery (WT) perform better overall. Among these, I-R structure alignment performs least well. This performance might be improved by optimising the alignment scores for our dataset; the alignment weights were trained on RISM incipits and might therefore not fit the folk songs optimally.

Wavelet transform seems to capture some essential notions of music similarity for finding correct occurrences, showing that essentially the same technique - fixed-length comparison with city-block distance - can be much more successful if it is applied to a different abstraction level than pitch sequences. Possibly a variable-length comparison step would yield even better results.

As expected from its success in symbolic melodic similarity MIREX tracks, B-spline alignment successfully retrieves a large portion of relevant occurrences annotated by human experts. However, it does not perform as well as some of the other measures in our comparison.

Confirming earlier research on melodic similarity in folk songs, alignment performs well in our task. We show that local alignment is very successful in correctly identifying

occurrences, even with a very simple substitution score, which only rewards equal pitches. Even better results might be achieved with different weights and substitution scores.

SIAM, to our knowledge, has not been evaluated for detecting phrase occurrences in folk song melodies yet, but performs on the same level as local alignment. This implies that SIAM is a good candidate for finding occurrences of melodic segments successfully, especially in corpora where transposition differences cannot be resolved through pitch histogram intersection, for instance in classical music and jazz, where key changes might make the estimation of transposition more difficult.

With maximal F1-scores of 0.73, the results of local alignment and SIAM come close to the between-annotator F1-scores between 0.75 and 0.78. This shows that we cannot do much better for our problem on this dataset without overfitting.

7. CONCLUSION

We conclude that both local alignment and SIAM seem adequate methods for finding occurrences of melodic segments in folk songs. Based on the retrieval scores, they find almost the same amount of relevant occurrences as human annotators among each other.

The measures investigated in this paper were applied to specific music representations. A wider range of music representations will be compared in future work. Moreover, the results will need to be analyzed in more detail with special attention to the cases where the similarity measures err, i.e. are false positives and false negatives more frequent for a specific tune family? And if so, do the annotators also disagree most on these same tune families? Besides, it is important to investigate the true positives as well, and ascertain that they are found in the correct positions in a melody.

The similarity measures compared in this article can be applied to other music corpora, which will give even deeper insights into relationships between melodies based on melodic segments that are shared between them. We can learn much about melodic identity and music similarity from both the confirmation and refutation of our findings in other music genres.

8. ACKNOWLEDGEMENTS

Berit Janssen and Peter van Kranenburg are supported by the Computational Humanities Programme of the Royal Netherlands Academy of Arts and Sciences, under the auspices of the Tunes&Tales project. For further information, see <http://ehumanities.nl>. Anja Volk is supported by the Netherlands Organisation for Scientific Research through an NWO-VIDI grant (276-35-001). We thank Gisela Velarde, Maarten Grachten and Julián Urbano for kindly providing their code and helpful comments, Sanneke van der Ouw, Jörn Janssen and Ellen van der Grijn for their annotations, and the anonymous reviewers for their detailed suggestions.

9. REFERENCES

- [1] Samuel P. Bayard. Prolegomena to a Study of the Principal Melodic Families of British-American Folk Song. *The Journal of American Folklore*, 63(247):1-44, 1950.
- [2] Michael Scott Cuthbert and Christopher Ariza. music21 : A Toolkit for Computer-Aided Musicology and Symbolic Music Data. In *11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, number Ismir, pages 637-642, 2010.
- [3] Elizabeth R. Delong, David M. Delong, and Daniel L. Clarke-Pearson. Comparing the Areas Under Two or More Correlated Receiver Operating Characteristic Curves: A Nonparametric Approach. *Biometrics*, 44(3):837-845, 1988.
- [4] Maarten Grachten, Josep Lluís Arcos, and Ramon López de Mántaras. Melody Retrieval using the Implication / Realization Model. In *MIREX-ISMIR 2005: 6th International Conference on Music Information retrieval*, 2005.
- [5] Peter Van Kranenburg, Martine De Bruin, Louis P Grijp, and Frans Wiering. The Meertens Tune Collections. Technical report, Meertens Online Reports, Amsterdam, 2014.
- [6] David Meredith. COSIATEC and SIATECCompress: Pattern Discovery by Geometric Compression. In *Music Information Retrieval Evaluation eXchange*, 2014.
- [7] David Meredith, Kjell Lemström, and Geraint A. Wiggins. Algorithms for discovering repeated patterns in multidimensional representations of polyphonic music. *Journal of New Music Research*, 31(4):321-345, 2002.
- [8] Marcel Mongeau and David Sankoff. Comparison of Musical Sequences. *Computers and the Humanities*, 24:161-175, 1990.
- [9] Eugene Narmour. *The Analysis and Cognition of Basic Melodic Structures. The Implication-Realization Model*. University of Chicago Press, Chicago, 1990.
- [10] Travis E. Oliphant. Python for Scientific Computing. *Computing in Science and Engineering*, 9(3):10-20, 2007.
- [11] Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-charles Sanchez, and Markus Müller. pROC : an open-source package for R and S + to analyze and compare ROC curves. *BMC Bioinformatics*, 12(1):77, 2011.
- [12] Deborah K. Scherrer and Philip H. Scherrer. An Experiment in the Computer Measurement of Melodic Variation in Folksong. *The Journal of American Folklore*, 84(332):230-241, 1971.
- [13] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195 - 197, 1981.
- [14] Wolfram Steinbeck. *Struktur und Ähnlichkeit. Methoden automatisierter Melodienanalyse*. Bärenreiter, Kassel, 1982.
- [15] Julián Urbano, Juan Lloréns, Jorge Morato, and Sonia Sánchez-Cuadrado. MIREX 2012 Symbolic Melodic Similarity: Hybrid Sequence Alignment with Geometric Representations. In *Music Information Retrieval Evaluation eXchange*, pages 3-6, 2012.
- [16] Peter van Kranenburg, Anja Volk, and Frans Wiering. A Comparison between Global and Local Features for Computational Classification of Folk Song Melodies. *Journal of New Music Research*, 42(1):1-18, 2012.
- [17] Gissel Velarde and David Meredith. A Wavelet-Based Approach to the Discovery of Themes and Sections in Monophonic Melodies. In *Music Information Retrieval Evaluation eXchange*, 2014.
- [18] Gissel Velarde, Tillman Weyde, and David Meredith. An approach to melodic segmentation and classification based on filtering with the Haar-wavelet. *Journal of New Music Research*, 42(4):325-345, December 2013.

PAD AND SAD: TWO AWARENESS-WEIGHTED RHYTHMIC SIMILARITY DISTANCES

Daniel Gómez-Marín **Sergi Jordà** **Perfecto Herrera**
Universitat Pompeu Fabra Universitat Pompeu Fabra Universitat Pompeu Fabra
daniel.gomez@upf.edu sergi.jorda@upf.edu perfecto.herrera@upf.edu

ABSTRACT

Measuring rhythm similarity is relevant for the analysis and generation of music. Existing similarity metrics tend to consider our perception of rhythms as being in time without discriminating the importance of some regions over others. In a previously reported experiment we observed that measures of similarity may differ given the presence or absence of a pulse inducing sound and the importance of those measures is not constant along the pattern. These results are now reinterpreted by refining the previously proposed metrics. We consider that the perceptual contribution of each beat to the measured similarity is non-homogeneous but might indeed depend on the temporal positions of the beat along the bar. We show that with these improvements, the correlation between the previously evaluated experimental similarity and predictions based on our metrics increases substantially. We conclude by discussing a possible new methodology for evaluating rhythmic similarity between audio loops.

1. INTRODUCTION

Rhythm similarity is an important problem for both music cognition and music retrieval. Determining which aspects of the musical flow are used by musical brains to decide if two musical excerpts share similarities with respect to rhythm, would make it possible to build algorithms that approximate human ratings about such relatedness. The applications of such algorithms in MIR contexts should be obvious and some have already been addressed [33] [13] [6] [20]. Unfortunately, there is a gap between the knowledge provided by cognitive models and engineering models with respect to similarity in general, and rhythm similarity in particular. Rhythm similarity metrics used in MIR are frequently based on superficial information such as inter-onset intervals, overall tempo or beat rate, onset density, and they usually consider full-length songs to derive a single similarity value. Contrastingly, rhythm similarity models developed by cognitive scientists insist on the importance of syncopation, beat salience, periodicities


and shorter time-scales to determine similarity. In this paper we address the above-mentioned gap and propose two rhythm similarity distances that refine those currently available (and probably rougher than desirable). The proposed distances have been derived from music cognition knowledge and have been tuned using experiments involving human listeners. We additionally show that they can be adapted to work (at least) in a music-loop collection organization context, where music creators want to organize their building blocks in rhythm-contrasting or rhythm flowing ways where similarity would provide the criterion for such concatenation of elements.

Previous work has used rhythmic descriptors, computed from audio signals, to analyze song databases. A common collection used for testing genre classification methodologies, The Ballroom Dataset, has been sorted automatically using different rhythmic descriptors and methodologies [4] [29] [9] [24]. Out of the ballroom dataset very few authors have addressed rhythm in electronic music with rhythmic descriptors [10] [23] [2]. The logic behind most of these research is the assumption that if a corpus is classified according to annotated labels, the features used for that clustering are somehow related to the phenomena that generate the clustering. In other words, a correct classification implies that the features used are useful despite their perceptual relevance.

Using symbolic representations of music, other authors propose metrics to evaluate rhythmic similarity that have shown to be useful in melody classification [33] or have proven correlation with cognitive judgements in rhythmic similarity experiments [12] [25] [1].

However, neither the audio-based methodologies or the symbolic metrics for rhythm similarity ([23] being an exception) have been designed for exploring short audio segments such as loops. Moreover, methodologies to evaluate rhythmic similarity between two audio loops and retrieve a value that can be analogous to a human rating are not yet available. Therefore we want to develop perceptually grounded rhythm similarity metrics to be used with short audio loops.

This paper is aimed to present two new rhythmic similarity metrics derived from revisiting the results of our cognitive experiments on rhythm similarity perception [8]. After revisiting our previous experiments, two metrics arise as useful in similarity prediction tasks. Based on those metrics we then introduce a new methodology to explore rhythmic similarity between audio loops.

 © Daniel Gómez-Marín, Sergi Jordà, Perfecto Herrera. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Daniel Gómez-Marín, Sergi Jordà, Perfecto Herrera. "PAD and SAD: Two Awareness-Weighted Rhythmic Similarity Distances", 16th International Society for Music Information Retrieval Conference, 2015.

The metrics proposed are based on the requirement that rhythmic similarity must be rooted in current knowledge of rhythm perception, where the notions of beat entrainment, reinforcement and syncopation are fundamental. We hypothesize that a proper rhythmic similarity measure can be built upon those perceptual considerations, emphasizing the idea that our attention when judging the similarity between two rhythms is not evenly distributed in time. We specifically propose that we are more aware of certain regions of a rhythm than others, affecting the way in which we measure their similarity. To test our hypothesis we use the results of our previous perceptual experiments and compare them with predictions computed with our metrics for the same rhythmic patterns in order to determine their correlation.

High correlation values between the similarity ratings of our previous experiment and the metrics presented here are found, suggesting that blending awareness and syncopation is important for accurately predicting rhythmic similarity. Finally we want to explore if the measures we propose, besides providing good fits and predictions of human judgements, can be used to organize loop collections. The use of our metrics in audio analysis will be discussed in the last sections of the paper, where we propose a methodology and evaluate it using audio loops of drum break patterns. Our results for this pilot validation present significant correlations between the similarity judgements of the subjects and the predicted distances proposed here.

2. STATE OF THE ART

2.1 Beat Induction

The fact that us humans induce a pulse sensation when listening to music is by no means trivial and it seems to be an innate and involuntary process [34]. It is known that the mechanisms that favour our acquisition of a beat when listening to music can also be triggered by any sequence of onsets [26]. This emergent beat entrainment is a cognitive process that can be divided two stages: first, we try to infer a metrical structure either by computing distances from intervals of the musical surface, where at least 5 to 10 notes are needed [3], or just try to match the incoming sound to an internal repertoire of known rhythms. Finally, once a meter has been hypothesized, it is maintained in the form of expectancies that interact with the new incoming sounds [17]. During this interaction, the expected pulse can be reinforced or disconfirmed. When challenged, brain rejection signals have been measured by means of EEG [15]. The occurrence of a disconfirmation is often referred to as syncopation, indicating notes that were expected on the beat but were presented on a previous metrical position [18].

In order to represent the variability of expectancies along a rhythmic pattern, researchers use profiles that indicate the metrical weight of a note depending on its position. Different profiles that highlight the importance of a beat reinforcement or a syncopated event, depending on its occurrence within a full metrical period, have been proposed.

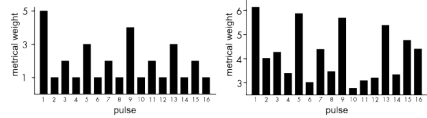


Figure 1. Lerdhal and Jackendorff's [16] metrical weight profile (left) and the experimentally revised version of Palmer and Krumhansl [22] measured for musicians (right).

A main theoretical profile [16] and an updated version experimentally revised with musicians [22] are presented in Figure 1. These profiles stress the existence of a perceptual hierarchy of sound events depending on their occurrence, suggesting that some reinforcements or syncopations are perceptually more relevant than others. These ideas have led to algorithms that measure the syncopation of a monomimbral unaccented phrase [30]. Moreover, these algorithms have been used to correlate syncopation with the difficulty to tap along rhythms [5], musical complexity [31] [7] [27] and musical pleasure and desire to dance [35], stressing the idea that syncopation has a powerful effect on our perception of music.

2.2 Rhythmic Similarity

Once we can extract a numerical value from a pattern of onsets such as its syncopation value, comparing patterns and establishing distances between them is mathematically possible. One main approach, proposed by Johnson-Laird [14], is to analyze the onsets present on every beat of a rhythmic pattern and assign the beat to a category depending if it reinforces the beat, challenges the beat or does nothing to the beat. This approach has been modified [28] and successfully tested with humans under experimental conditions [1]. These ideas will be further expanded throughout this paper.

As most proposed similarity metrics are measured on monomimbral, monotonal and unaccented symbolic representations of rhythm, there are others who have explored the use of string similarity techniques as the swap distance or the edit distance [19] [21] to measure similarity between patterns. The edit distance has proven to be a useful predictor of human similarity judgements [32] [11] [25]. But still, the obtained fit between the edit distance and subjective similarity judgments has a big room for improvement.

Here we use similarity metrics based on syncopation, specifically a variation of the theory of Johnson-Laird in which we expand the possible groups that a beat can be subscribed to (syncopation, reinforcement or nothing). In the following sections we present, test and discuss an improvement over a previously published metric and explore the possibility of using these symbolic metrics in rhythmic analysis of audio signals.

3. METHOD

In this section we present different concepts that are the building blocks of our rhythmic similarity algorithms. We have to make some simplifying assumptions, considering one bar, monotimbral, monotonal, percussive patterns with 4/4 time signature and a minimum resolution of a sixteenth note. The symbolic representation of such patterns is binary, where a 1 indicates an onset and 0 indicates a silence. Therefore the patterns used throughout this work are 16 digit sequences of zeroes and ones.

3.1 Beats to syncopation groups

Rhythms are split in beats, in our case each beat has four steps (four digits). Each beat of a rhythm is classified into a group according to its relation with the pulse, either a reinforcement or a challenge (See table 1). This method is a variation of Johnson-Laird's method [14], in which beats are clustered in three broad categories: syncopation, reinforcement or nothing depending if the elements of the beat are a reinforcement, a challenge or have no interaction with the pulse. We have expanded Johnson-Laird's method by splitting syncopation into three possible groups (groups 5 to 7, Table 1), reinforcement is split in three groups (groups 1 to 3, Table 1) and adding a new category where a syncopation and a reinforcement are both present (group 8, Table 1). Expanding the groups in which a beat can be classified offers more detail on the role of each segment and helps differentiate between different syncopations or different reinforcements.

The procedure to classify each beat is to compute its syncopation value using the beat profile 2 0 1 0. This profile is derived from Lerdahl and Jackendorff's [16] in which weights are proportional to the duration of the note each accent represents: an accent of a whole note has a higher weight than an accent on a half note, which is higher than an accent on a quarter note, and so forth. In our beat profile the first onset, that is coincident with the pulse, has a higher weight than the third onset which is coincident with an eighth note.

It is important to note that an onset on the fourth step of a beat generates a syncopation only if the first step of the next beat is a silence. Therefore to calculate the appropriate syncopation values for every beat, the first step of the following beat has to be considered. The syncopation value for each beat is the sum of each onset's metrical weights.

Each beat can then be assigned to one out of eight syncopation categories, but we have considered the case of a reinforcement on the first step and a syncopation on the fourth step 1001_ (total syncopation value = 0) as special cases belonging to syncopation group #8.

3.2 Coincidence

We propose here two metrics, one that explores if two patterns have the same onsets and silences on a specific beat, which we call pattern coincidence distance (PD) and the other one, named syncopation coincidence distance (SD)

Group	value	Patterns
1	3	1010_ 1010x
2	2	1000_1000x 1001x 1011x
3	1	0010_ 0010x 0110_ 0110x 1110_ 1110x
4	0	0000_ 0000x 1111x 0011x 0001x 0111x
5	-1	0100_ 0100x 1100_ 1100x 0101x 1101x
6	-2	0001_ 0011_ 0111_ 1111_
7	-3	0101_ 1101_
8	0	1001_ 1011_

Table 1. Relation between syncopation group, syncopation value and beat patterns. The symbol '_' indicates a silence at the beginning of the next beat and the symbol 'x' indicates an onset at the beginning of the next beat.

which explores if a specific beat of two patterns belong to the same syncopation group (see Table 1).

Here we give an illustrative example to understand PD and SD. The two first beats of a given pattern A have the following onset/silence configuration 1001 0110 and another pattern B has 1100 0010. Their respective syncopation groups are #8 #3 and #5 #3. The pattern coincidence (PD) is computed by looking at the percentage of coincident onsets and silences on the same beat of each pattern. Their coincidence values would be $(2+3)/8 = 0.625$ because for the first beat there are 2 out of four notes coincident between 1001_ and 1100; and for the second beat, there are 3 coincidences between 0110 and 0010. In total there are 2+3 coincidences out of 8 possible. On the other hand, to measure the syncopation coincidence (SD), for the first beat of patterns A and B, we get that 1001_ belongs to family #8 and 1100 belongs to family #5. Clearly 8 is different from 5. But if we look at the second beat, 0110 and 0010 belong to the same group #3, thus group coincidence is $0+1=1$. With these metrics we obtain two methods for measuring a numerical value of the coincidence between two coincident beats of different patterns. If the coincidence between all the beats of two patterns is computed, this value can be used as a measure of similarity between the two patterns. However, we might consider that, as different onsets have different metrical weights depending on their position within a pattern (see Figure 1), beats can also have different perceptual relevance depending on their position within the pattern. In this paper we have conceptualized this factor as awareness.

3.3 Awareness as an effect of metrical hierarchy

Our previously published results [8] suggest a difference in the relevance of each beat when measuring similarity between two patterns based on coincidence. This awareness has proven important when exploring correlations between our experimental results of similarity and the rhythmic patterns compared. Thus we propose each beat to have different relevance when evaluating similarity between two patterns in the presence of a pre defined metrical context. Awareness is conceived as weight factors applied to each beat's coincidence metric (either PD or SD). These weights

emphasize or moderate each beat's importance on the final distance value. This concept will be addressed in the following section and is decisive for explaining our results.

3.4 Rhythmic Similarity Metrics

Our metrics are straightforward and are based on computing any of the two types of coincidence (either beat or syncopation group), and using them directly or with an awareness-based weighting. We finally have four metrics, two of them non-weighted. Pattern coincidence Distance (PD) and Syncopation group coincidence Distance (SD), Pattern coincidence and Awareness Distance (PAD) and Syncopation group coincidence and Awareness Distance (SAD). The weights of the PAD and SAD metrics will be explored in the following sections.

$$PD = pc1 + pc2 + pc3 + pc4 \quad (1)$$

$$SD = sc1 + sc2 + sc3 + sc4 \quad (2)$$

$$PAD = pc1w1 + pc2w2 + pc3w3 + pc4w4 \quad (3)$$

$$SAD = sc1w1 + sc2w2 + sc3w3 + sc4w4 \quad (4)$$

Where $pc(n)$ is pattern coincidence, $sc(n)$ is syncopation group coincidence, $w(n)$ is the weighting of each beat, n is the order of the beat within a full metric cycle.

4. EXPERIMENT

In previously published paper [8] we performed two rhythmic similarity experiments, one inducing the beat and another without inducing the beat. In this paper we are revisiting the beat-induced experiment to test our new metrics with the similarity ratings obtained in the previous one.

In one of the experiments, twenty one subjects (recruited among the MTG staff and UPF pool of students, all of them with musical experience of more than 5 years as amateur performers) rated different rhythm pairs in the presence of a beat-inducing kick drum. The rhythm pairs were constructed by making variations of a main pattern as shown in Table 2. A region of the base pattern was progressively shifted, generating new patterns. Nine different main patterns were designed and the length and origin of the region was varied systematically. Thirty six rhythm pairs plus a control pair were tested by all the subjects who rated similarity using a Likert scale of seven steps. To promote rhythm entrainment, a kick drum, coincident with the start of every beat, was presented before and simultaneously with the tested rhythms.

5. RESULTS

The mode of the similarity ratings for each pair of patterns was used as the value capturing their similarity. All the pairs of patterns presented to the subjects are analyzed with the metrics described in section 3, exploring the correlations with the similarity ratings reported for each pair.

In our previously reported experiment, we computed the PD distance for every tested pair and observed a Spearman Rank correlation with the subject's similarity ratings

Base Pattern	variation
1010 1110 1000 1010	1101 0110 1000 1010
1010 1110 1000 1010	1010 1011 1000 1010
1010 1110 1000 1010	1001 0101 1000 1010
1010 1110 1000 1010	1010 1010 1100 1010

Table 2. Example of four stimuli pairs used in the experiment. The left column has the base pattern and the derived variations are on the right column. The similarity measures of the subjects are between the base pattern and each variation. The underlined portion of the base pattern is repeated in the variations.

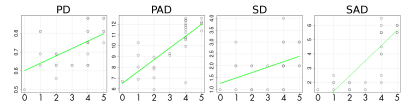


Figure 2. PD, PAD, SD and SAD predictions correlated with similarity ratings. X axis: similarity ratings, y axis PD, PAD, SD and SAD predictions from left to right.

of 0.54 (p-value < 0.005). We also computed the SD distance which has a Spearman rank correlation value of 0.46 with the similarity ratings (p-value < 0.01).

Here we calculate our newly introduced metrics PAD and SAD (see Figure 2). To calculate PAD, a linear regression between the coincidence result of each beat and the similarity ratings is computed. The normalized weights obtained for beats 1 to 4 are 1, 0.27, 0.22 and 0.16 respectively. We take the weights of the linear regression as indications of the awareness for each beat. Using those weights we get the PAD distance with a Spearman Rank correlation value of 0.76 (p-value < 0.001). To calculate SAD a linear regression between each beats coincidence and similarity ratings generated the following normalized weights for beat 1 to 4: 1, 0.075, 0.14 and 0.12 respectively. Again, we take the weights of the linear regression and use them as indications of the awareness for each beat. Applying those weights we get the SAD distance which has a Spearman Rank correlation value of 0.81 (p-value < 0.001).

The resulting awareness profiles of both PAD and SAD metrics have a similar behaviour (see Figure 3). In both cases the importance of the first beat is almost 5 times larger than the other beats. Our experimental hypothesis is that this phenomena evidences a hierarchical organization of rhythmic elements in time where the first element of a rhythmic sequence is of greater importance than the rest.

The correlation values that have been obtained suggest that the PAD and SAD metrics are better than previously existing candidates to predict rhythmic similarity between two patterns of onsets in the presence of a beat, the way in which most of the music is experienced. The PAD and SAD metrics surpass the results found and reported in our previous experiment, which makes them suitable to be used

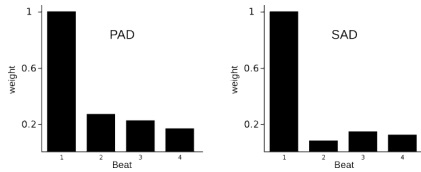


Figure 3. Awareness profiles of the PAD and SAD distances that generated best correlations with rhythm similarity ratings.

in real life scenarios.

6. DISCUSSION

It can be seen that the SAD metric has the highest correlation values with human similarity, rating slightly above the PAD metric, while the non-weighted metrics PD and SD are significantly lower. This suggests that the concepts of syncopation groups and beat awareness are perceptually relevant.

The drop in correlation values when there is no awareness weighting validates the idea of each beat having a different importance when beat induced subjects try to make sense of them. It seems that the first beat is the most important followed by the third, the fourth and the second.

The SAD metric is based on comparing if syncopation groups are coincident between different patterns (see section 3.2). This means that a change from one family to any other family is penalized by our algorithm despite if the change is between syncopation to syncopation (groups 5 to 7 in Table 1) or reinforcement to reinforcement (groups 1 to 3 in Table 1) or if it is a change from a syncopation to a reinforcement group or to the nothing group (or vice versa). Since the SAD metric has a positive correlation with similarity ratings, this suggests that any change between groups decreases our perception of similarity. On the other hand, perception of rhythmic similarity is highly influenced with the coincidence between syncopation groups or patterns and the position of those coincidences within the pattern.

7. PILOT VALIDATION

A straightforward application of PAD and SAD for exploring rhythm-similarity-based loop exploration can be discussed. The simplest approach would be to use an onset detector to the loop signal and extract a general onset pattern. This would lead to a single-level pattern deprived of any instrumental information where all musical interplay, the main information, would be lost. On the other hand, a robust source separation system would be ideal, where an audio loop could be completely split into its different instrumental components and then converted to a symbolic representation. But the technologies to perform such a task are not yet reliable. An alternative would be to extract onset patterns from meaningful frequency bands that could

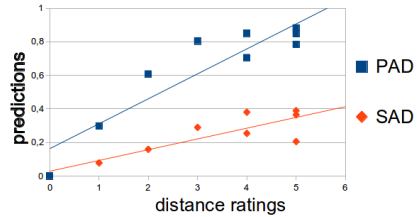


Figure 4. Predicted similarity vs similarity ratings of ten audio loops using our methodology with PAD and SAD metrics.

preserve spectral information present on the audio loop.

We propose a methodology where a sound loop, of known metric length, is segmented every sixteenth note value and filtered in 23 Bark bands. This is a typical spectral representation which approximates frequency resolution of the human hearing. The energy peaks in each band are considered as onsets and the rest as silences. In this way we convert an audio loop into a binary matrix of onset and silences of 23 bands times the number of analysis windows. An audio loop is then decomposed in 23 parallel rhythmic patterns that can be compared with the 23 patterns of another audio loop measuring PAD and SAD distances between bands. The sum of the band to band distances is the overall PAD or SAD distance between two audio loops. Note that this methodology is tempo independent if the loops compared have the same known metrical length.

As a pilot validation for our methodology, an experiment was carried out using nine different drum break loops in audio format (downloaded from <http://rhythm-lab.com>). All loops were post processed to have a metrical length of two bars. Fifteen musically trained subjects were invited to rate the rhythmic similarity between one audio loop and the rest using a Likert scale divided in 5 steps, from "very similar" to "very different". The mode of the results for each pair was used as the representative similarity value and the correlations with PAD and SAD distances were measured. The awareness profile used for both PAD and SAD was 1 0.075 0.14 0.12 extracted from the results presented in section 5 (see Figure 3, right).

The obtained results present (p-value < 0.001) a significant correlation between the similarity reported by the fifteen subjects and the PAD and SAD distances (Figure 4). The PAD distance has a 0.80 Spearman rank correlation value (p-value < 0.01). The SAD distance has a Spearman correlation value of 0.75 (p-value < 0.05).

It is quite interesting that PAD and SAD distances provide reliable similarity predictions, given the subjectivity of the task and the fact that the breaks come from very different recordings with an obvious difference in timbre and dynamics. For this pilot validation The PAD has a higher correlation value with the similarity ratings than the SAD metric.

8. CONCLUSION AND FUTURE WORK

Based on these results, we propose that measuring the PAD and SAD distance between two rhythms with an induced beat as metrical context provider, is an effective way to predict human rhythmic similarity ratings. Perceptually motivated rhythm similarity measures that are applied to MIR problems should take into account both the syncopation groups and a beat-awareness measure, in order to match subjective appreciations of rhythm similarity.

The rhythms used in the foundational experiments of our metrics are only limited to a 4/4 time signature, a 16 step length, sixteenth note resolution and binary dynamics. Expanding the signature to other common signatures, smaller note resolutions and subtler dynamics is important in order to broaden the validity and usefulness of our metrics and methodology.

Even though our methodology for measuring similarity among loops yielded significant high correlation values, both with PAD and SAD, it is important to consider the scale of the pilot validation is limited. New experiments with a higher amount of loops should be carried out in order to explore the real advantages and limitations of our methodology.

9. ACKNOWLEDGEMENTS

We would like to thank Julián Burbano for his help in the analysis of the data. This research has been partially supported by the EU funded GiantSteps project (FP7-ICT-2013-10 Grant agreement nr 610591).

10. REFERENCES

- [1] Erica Cao, Max Lotstein, and Philip N. Johnson-Laird. similarity and families of musical rhythms. *Music Perception: An Interdisciplinary Journal*, 31(5):444–469, 2014.
- [2] Nick Collins. Influence in early electronic dance music: an audio content analysis investigation. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR 2012)*, number Ismir, pages 1–6, 2012.
- [3] Peter Desain and Henkjan Honing. Computational models of beat induction: The rule-based approach. *Journal of New Music Research*, 28(1):29–42, 1999.
- [4] Simon Dixon, Fabien Gouyon, and Gerhard Widmer. Towards characterisation of music via rhythmic patterns. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR 2004)*, volume 5, pages 509–516, 2004.
- [5] W. Tecumseh Fitch and Andrew J. Rosenfeld. Perception and Production of Syncopated Rhythms. *Music Perception*, 25(1):43–58, 2007.
- [6] Jamie Forth. *Cognitively-motivated geometric methods of pattern discovery and models of similarity in music*. PhD thesis, Goldsmiths, University of London, 2012.
- [7] Francisco Gómez, Eric Thul, and Godfried T. Toussaint. An experimental comparison of formal measures of rhythmic syncopation. *Proceedings of the International Computer Music Conference*, pages 101–104, 2007.
- [8] Daniel Gómez-Marín, Sergi Jordà, and Perfecto Herrera. Strictly Rhythm: Exploring the effects of identical regions and meter induction in rhythmic similarity perception. In *11th International Symposium on Computer Music Multidisciplinary Research (CMMR)*, Plymouth, 2015.
- [9] Fabien Gouyon, Simon Dixon, Elias Pampalk, and Gerhard Widmer. Evaluating rhythmic descriptors for musical genre classification. In *AES 25th International Conference*, pages 1–9, LONDON, 2004.
- [10] Matthias Grühne, Christian Dittmar, and Daniel Gaertner. Improving Rhythmic Similarity Computation by Beat Histogram Transformations. *10th International Society for Music Information Retrieval Conference (ISMIR 2009)*, (Ismir):177–182, 2009.
- [11] Catherine Guastavino, Francisco Gómez, Godfried T. Toussaint, Fabrice Marandola, and Emilia Gómez. Measuring Similarity between Flamenco Rhythmic Patterns, 2009.
- [12] Ludger Hofmann-Engl. Rhythmic similarity: A theoretical and empirical approach. *7th International Conference on Music Perception and Cognition*, (c):564–567, 2002.
- [13] Henkjan Honing. Lure(d) into listening: The potential of cognition-based music information retrieval. *Empirical Musicology Review*, 5(4):121–126, 2010.
- [14] Philip N. Johnson-Laird. Rhythm and meter: A theory at the computational level. *Psychomusicology: A Journal of Research in Music Cognition*, 10(2):88–106, 1991.
- [15] Olivia Ladinig, Henkjan Honing, Gábor Háden, and István Winkler. Probing Attentive and Preattentive Emergent Meter in Adult Listeners without Extensive Music Training, 2009.
- [16] Fred Lerdahl and Ray Jackendoff. *A generative theory of tonal music*. MIT Press, 1985.
- [17] Justin London. *Hearing in Time*. Oxford University Press, 2012.
- [18] H. Christopher Longuet-Higgins and Christopher S. Lee. The Rhythmic Interpretation of Monophonic Music. *Music Perception: An Interdisciplinary Journal*, 1(4):424–441, 1984.
- [19] Marcel Mongeau and David Sankoff. Comparison of musical sequences. *Computers and the Humanities*, 24(3):161–175, 1990.

- [20] Alberto Novello, Martin M.F. McKinney, and Armin Kohlrausch. Perceptual Evaluation of Inter-song Similarity in Western Popular Music. *Journal of New Music Research*, 40(1):1–26, 2011.
- [21] Keith S. Orpen and David Huron. Measurement of similarity in music: A quantitative approach for non-parametric representations. *Computers in music research*, 4:1–44, 1992.
- [22] Caroline Palmer and Carol L. Krumhansl. Mental representations for musical meter. *Journal of experimental psychology. Human perception and performance*, 16(4):728–741, 1990.
- [23] Maria Panteli, Bruno Rocha, Niels Bogaards, and Aline Honingh. Development of a Rhythm Similarity Model for Electronic Dance Music. Number Ismir, pages 537–542, 2014.
- [24] Geoffroy Peeters. Spectral and Temporal Periodicity Representations of Rhythm for the Automatic Classification of Music Audio Signal. *IEEE Transactions on Audio, Speech, and Language Processing*, 19(5):1242–1252, 2011.
- [25] Olaf Post and Godfried T. Toussaint. The Edit Distance as a Measure of Perceived Rhythmic Similarity. *Empirical Musicology Review*, 6(3):164–179, 2011.
- [26] Dirk-jan Povel and Peter Essens. Perception of Temporal Patterns. *Music Perception*1, 2(4):411–440, 1985.
- [27] Jeffrey Pressing. Cognitive complexity and the structure of musical patterns. *Noetica*, 3(8):1–8, 1998.
- [28] Jasba Simpson and David Huron. The Perception of Rhythmic Similarity: A Test of a Modified Version of Johnson-Lairds Theory. *Canadian Acoustics*, 21(3):89–94, 1993.
- [29] Leigh M. Smith. Rhythmic similarity using metrical profile matching. In *International Computer Music Conference*, 2010.
- [30] Leigh M. Smith and Henkjan Honing. Evaluating and extending computational models of rhythmic syncopation in music. In *Proceedings of the International Computer Music Conference*, pages 688–691, 2006.
- [31] Eric Thul and Godfried T. Toussaint. Rhythm complexity measures: a comparison of mathematical models of human perception and performance. In *Proceedings of the International Society for Music Information Retrieval (ISMIR) Conference, Section 5c - Rhythm and Meter*, number 8, pages 663–668, 2008.
- [32] Godfried T. Toussaint, Malcolm Campbell, and Naor Brown. Computational Models of Symbolic Rhythm Similarity: Correlation with Human Judgments. *Analytical Approaches to World Music*, 1(2), 2011.
- [33] Anja Volk, Jörg Garbers, Peter Van Kranenburg, Frans Wiering, Remco C Veltkamp, and Louis P Grijp. Applying rhythmic similarity based on inner metric analysis to folksong research. In *Eighth International Conference on Music Information Retrieval. Austrian Computer Society*, 2007.
- [34] István Winkler, Gábor P Hádén, Olivia Ladinig, István Sziller, and Henkjan Honing. Newborn infants detect the beat in music. *Proceedings of the National Academy of Sciences of the United States of America*, 106(7):2468–2471, 2009.
- [35] Maria A G Witek, Eric F. Clarke, Morten L. Kringelbach, and Peter Vuust. Effects of Polyphonic Context, Instrumentation, and Metrical Location on Syncopation in Music. *Music Perception: An Interdisciplinary Journal*, 32(2):201 – 217, 2014.

FOUR TIMELY INSIGHTS ON AUTOMATIC CHORD ESTIMATION

Eric J. Humphrey^{1,2} and Juan P. Bello¹

¹Music and Audio Research Laboratory, New York University

²MuseAmi, Inc.

ABSTRACT

Automatic chord estimation (ACE) is a hallmark research topic in content-based music informatics, but like many other tasks, system performance appears to be converging to yet another glass ceiling. Looking toward trends in other machine perception domains, one might conclude that complex, data-driven methods have the potential to significantly advance the state of the art. Two recent efforts did exactly this for large-vocabulary ACE, but despite arguably achieving some of the highest results to date, both approaches plateau well short of having solved the problem. Therefore, this work explores the behavior of these two high performing, systems as a means of understanding obstacles and limitations in chord estimation, arriving at four critical observations: one, music recordings that invalidate tacit assumptions about harmony and tonality result in erroneous and even misleading performance; two, standard lexicons and comparison methods struggle to reflect the natural relationships between chords; three, conventional approaches conflate the competing goals of recognition and transcription to some undefined degree; and four, the perception of chords in real music can be highly subjective, making the very notion of “ground truth” annotations tenuous. Synthesizing these observations, this paper offers possible remedies going forward, and concludes with some perspectives on the future of both ACE research and the field at large.

1. INTRODUCTION

Among the various subtopics in content-based music informatics, automatic chord estimation (ACE) has matured into a classic MIR challenge, receiving healthy attention from the research community for the better part of two decades. Complementing our natural sense of academic intrigue, the general music learning public places a high demand on chord-based representations of popular music, as evidenced by large online communities surrounding websites like e-chords¹ or Ultimate Guitar². Given

the prerequisite skill necessary to manually identify chords from recorded audio, there is considerable motivation to develop automated systems capable of reliably performing this task.

The goal of ACE research is —or, at least, has been— to develop systems that produce “good” time-aligned sequence of chords from a given music signal. Supplemented by efforts in data curation [2], syntax standardization [8], and evaluation [13], the bulk of chord estimation research has concentrated on building better systems, mostly converging to a common architecture [4]: first, harmonic features, referred to as pitch class profiles (PCP) or *chroma*, are extracted from short-time observations of the audio signal [7]; these features may then be processed by any number of means, referred to in the literature as *pre-filtering*; next, *pattern matching* is performed independently over observations to measure the similarity between the signal and a set of pre-defined chord classes, yielding a time-varying likelihood; and finally, *post-filtering* is applied to this chord class posterior, resulting in a sequence of chord labels over time.

However, despite continued efforts to develop better features [11], more powerful classifiers [10], or advanced post-filtering methods [1], performance appears to be tapering off, as evidenced by recent years’ results at MIREX³. Thus, while other areas of machine perception, such as computer vision and speech recognition, are able to leverage modern advances in machine learning with remarkable success, two recent efforts in large vocabulary ACE were only able to realize modest improvements by comparison [3, 9]. Acknowledging this situation begs an obvious question: why is automatic chord estimation different, and what might be done about it? Through an investigation of system behaviour and detailed error analysis, the remainder of this paper is an effort to shed some light on the problem.

2. RESEARCH METHODOLOGY

2.1 Automatic Systems

Given its long history, there are ample potential automatic chord estimation systems that could be considered in this inquiry. Here, though, we choose to focus our investigation on two recent, data-driven, large vocabulary systems for which we are able to obtain software implementations,

^{*}Please direct correspondence to eric@museami.com

¹<http://www.e-chords.com>

²<http://www.ultimate-guitar.com>



© Eric J. Humphrey, Juan P. Bello.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Eric J. Humphrey, Juan P. Bello. “Four Timely Insights on Automatic Chord Estimation”, 16th International Society for Music Information Retrieval Conference, 2015.

³http://www.music-ir.org/mirex/wiki/MIREX/_HOME

providing control over training and choice of chord vocabulary. Additionally, these system architectures are quite different and should, as a result, yield different machine perspectives, a strategy that has proven useful in the analysis of beat tracking systems [15].

2.1.1 *K-stream GMM-HMM with Multiband Chroma*

The first system considered is a modern, high-performing GMM/HMM chord estimation system [3], referred to here as “kHMM.” A multiband chroma representation is computed from beat-synchronous audio analysis, producing four parallel feature representations. Each is modeled by a separate multivariate Gaussian Mixture Model (GMM), whereby all chroma vectors and chord labels are rotated to a C root. During inference, four separate observation likelihoods over all chord classes are obtained by circularly rotating the feature vector the GMM, thereby making the model transposition invariant. These four chord class posteriors are then decoded jointly using a k-stream HMM, resulting in a single beat-aligned chord sequence.

2.1.2 *Deep Convolutional Neural Network*

Acknowledging the recent widespread success of deep learning methods, a deep convolutional network is also considered [9], referred to as “DNN.” Time-frequency patches of local contrast normalized constant-Q spectra, on the order of one second, are transformed by a four-layer convolutional network. Finding inspiration in the root-invariance strategy of GMM training, explicit weight-tying is achieved at the classifier across roots such that all qualities develop the same internal representations, allowing the model to generalize to chords unseen during training. Following the lead of deep network research in automatic speech recognition, likelihood scaling is performed after training to control class bias resulting from the severe imbalance in the distribution of chords. Finally, chord posteriors are decoded via the Viterbi algorithm [5].

2.2 Evaluation

Expressed formally, the modern approach to scoring an ACE system is a weighted measure of chord-symbol recall, R_W , between a reference, \mathcal{R} , and estimated, \mathcal{E} , chord sequence as a continuous integral over time, summed over a discrete collection of N annotation pairs:

$$R_W = \frac{1}{S} \sum_{n=0}^{N-1} \int_{t=0}^{T_n} C(\mathcal{R}_n(t), \mathcal{E}_n(t)) dt \quad (1)$$

Here, C is a chord *comparison* function, bounded on $[0, 1]$, t is time, n the index of the track in a collection, T_n the duration of the n^{th} track. This total is normalized by the *support*, S , corresponding to the cumulative amount of time over which the comparison rule is defined for \mathcal{R} , given by the indicator function in a similar integral:

$$S = \sum_{n=0}^{N-1} \int_{t=0}^{T_n} \mathbb{1}_{\mathcal{R}_n(t)} dt \quad (2)$$

Defining the normalization term S separately is useful when comparing chord names, as it relaxes the assumption that the comparison function is defined everywhere. Furthermore, setting the comparison function as a free variable allows for flexible evaluation of a system’s outputs, and thus the focus on vocabulary can largely focus on the choice of comparison function, C . The work presented here leverages `mir_eval`, an open source evaluation tool-box providing a set of seven chord comparison functions, characterizing different relationships between chords [14].

2.3 Reference Annotations

2.3.1 *Ground Truth Data*

The first major effort to curate reference chord annotations, now part of the larger Isophonics⁴ dataset, covers the entire 180-song discography of *The Beatles*, as well as 20 songs from *Queen*, 14 from Carole King, and 18 from *Zweick*; due to content access, only the 200 songs from *The Beatles* and *Queen* are used here. Two other large chord annotation datasets were publicly released in 2011, offering a more diverse musical palette. The McGill *Billboard* dataset consists of over 1000 annotations, of which more than 700 have been made public. This project employed a rigorous sampling and annotation process, selecting songs from Billboard magazine’s “Hot 100” charts spanning more than three decades. The other, provided by the Music and Audio Research Lab (MARL) at NYU⁵, consists of 295 chord annotations performed by undergraduate music students; 195 tracks are drawn from the USPop dataset⁶, and 100 from the RWC-Pop collection⁷, in the hopes that leveraging common MIR datasets might facilitate access within the community. In all three cases, chord annotations are provided as “ground truth,” on the premise that the annotations represent the gold standard.

2.3.2 *The Rock Corpus*

Importantly, the reference chord annotations discussed previously offer a singular perspective, either as the output of one person or the result of a review process. The *Rock Corpus*, on the other hand, is a set of 200 popular rock tracks with time-aligned chord and melody transcriptions performed by two expert musicians [6]: one, a pianist, and the other, a guitarist, referred to as DT and TdC, respectively. This collection of chord transcriptions has seen little use in the ACE literature, as its initial release lacked timing data for the transcriptions. A subsequent release resolved this issue, however, and doubled the size of the collection. While previous efforts have sought to better understand the role of subjectivity in chord annotations [12], this dataset provides an opportunity to explore the behavior of ACE systems as a function of multiple reference transcriptions at a larger scale.

⁴<http://isophonics.net/content/reference-annotations>

⁵<https://github.com/tmc323/Chord-Annotations>

⁶<http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html>

⁷<https://staff.aist.go.jp/m.goto/RWC-MDB/rwc-mdb-p.html>

	Ref-DNN	Ref-kHMM	kHMM-DNN
root	0.789	0.808	0.840
thirds	0.757	0.775	0.815
majmin	0.759	0.776	0.798
mirex	0.769	0.783	0.806
triads	0.705	0.721	0.783
sevenths	0.620	0.645	0.691
tetrads	0.567	0.588	0.678
v157	0.649	0.659	0.678

Table 1. Weighted recall across comparison rules between the ground truth references and both models, respectively, as well as against each other.

3. LARGE-VOCABULARY CHORD ESTIMATION

Here we investigate large-vocabulary chord estimation as a basis for experimentation. First and foremost, it presents a particularly challenging problem, and therefore offers a good deal of potential for subsequent analysis. Large chord vocabularies also avoid the inherent noise introduced by approximately mapping chords into the classic major-minor formulation, e.g. $A:sus2 \rightarrow A:maj$ or $C:dim7 \rightarrow C:min$. Additionally, the large amount of available data should be sufficient for learning a large number of chord classes.

Before proceeding, the ground truth collections are merged for training and evaluation, totaling 1235 tracks. A total of 18 redundant songs are identified via the EchoNest Analyze API⁸ and removed to avoid potential data contamination during cross validation. All but one is dropped for each collision, preferring content from Iso-phonics, Billboard, and MARL, respectively, resulting in a final count of 1217 unique tracks.

To ensure a fair comparison between algorithms, the ground truth data is partitioned into five distinct splits. Training is repeated five times for both systems addressed in Section 2.1 for cross validation, such that each split is used as a holdout test set once. Both models adopt the same chord vocabulary, comprised of the thirteen most frequent chord qualities in all twelve pitch classes, as well as a no-chord class, for a total of 157 chord classes, consistent with previous efforts [3]. Chords outside this strict vocabulary are ignored during training, rather than mapped to their nearest class approximation. The Rock Corpus data is not used for training, and saved exclusively for analysis.

3.1 Experimental Results

Weighted recall is averaged over the five test splits are for all reference chord labels according to the seven *mir.eval* comparison rules, shown in Table 1. At first glance, the overall statistics seem to indicate that the two systems are roughly equivalent, with “kHMM” outperforming “DNN” by a small margin. The automatic systems perform best at root-level recall, and performance drops as the comparison rules encompass more chords. Notably, a comparison of algorithmic estimations, given in the third column, shows that these two systems do indeed offer very

⁸<http://developer.echonest.com/docs/v4>

	DT-TdC	(DT TdC)-DNN	(DT TdC)-kHMM
root	0.932	0.792	0.835
thirds	0.903	0.750	0.785
majmin	0.905	0.723	0.766
mirex	0.902	0.737	0.776
triads	0.898	0.719	0.760
sevenths	0.842	0.542	0.595
tetrads	0.835	0.540	0.590
v157	0.838	0.539	0.590

Table 2. Weighted recall across comparison rules for the two human annotators, and the better match of each against the two automatic systems.

different perspectives. Therefore, it will be valuable to not only investigate where the estimated chord sequences differ from the reference, but also how these estimated sequences differ from each other.

Similarly, weighted recall is also given for both systems over the Rock Corpus in Table 2. It is an open question as to how an estimated annotation might best be compared against more than one human reference. For the purposes of analysis, the best matching reference-estimation pair is chosen at the track level and used to compute the weighted average. Still, performance on the Rock Corpus is lower for both automatic algorithms. This is likely a result of a mismatch in chord vocabulary, as space of chords used in the Rock Corpus is a smaller subset than the 157 estimated by automatic systems. Additionally, it is curious to observe a non-negligible degree of disagreement between the two human perspectives, with more than a 15% discrepancy in the tetrads condition. That said, the human annotators do agree a deal more that is attained by either system, indicating that there is likely room for improvement.

3.2 Track-wise Visualizations

While weighted recall gives a good overall measure of system performance, we are particularly interested in developing a more nuanced understanding of how these systems behave. To this end, system performance is now examined at the track-level, as real music is often highly self-similar and the chords within a song will be strongly related. Errors and other kinds of noteworthy behavior should be well-localized as a result, making it easier to draw conclusions from the data.

Two track-wise scatter plots are given in Figure 1, for the ground truth and Rock Corpus datasets. The former compares the agreement between multiple *estimations*, along x , with the better matching estimation for the given reference, along y , where each quadrant characterizes a different behavior: (I), all annotations agree; (II), one estimation matches the reference better than the other; (III), all annotations disagree; and (IV), the estimations agree more with each other than the reference. Importantly, this track-wise comparison makes it easier to identify datapoints that can help address our original research questions. Tracks for which only one algorithm performs well (II) likely indicate boundary chords. Alternatively, instances where both algorithms produce poor estimations,

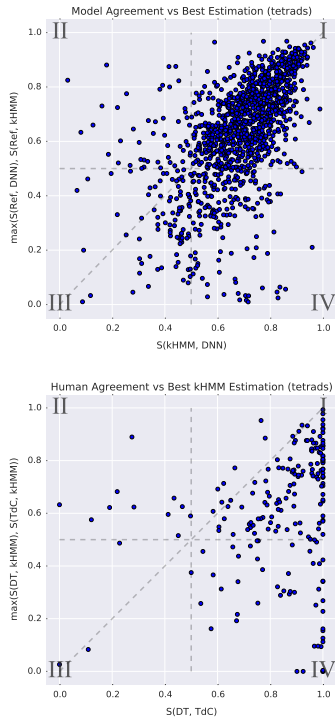


Figure 1. Trackwise recall for the “tetrads” in two conditions: (top) over the ground truth data, illustrating *model* agreement versus the better match between the reference and estimated annotations; (bottom) over the Rock Corpus data, illustrating annotator agreement versus the better match between the two reference and kHMM annotations.

and yet *neither* agree (III), are curious and warrant further inspection. Finally, tracks that result in similarly incorrect estimations (IV) highlight some kind of greater challenge to automatic systems.

The second plot, conversely, compares the agreement between multiple *references*, along x , with the better matching reference for the given estimation, along y , and analogous characterizations by quadrant: (I), all annotations agree; (II), one reference matches the estimation better than the other; (III), all annotations disagree; and (IV), the references agree more with each other than the estimation. Here, annotator disagreement in the presence of a matching estimation (II) is indicative of subjectivity, while disagreement between all annotations (III) is suspicious and should be explored. Furthermore, tracks with an es-

timated annotation that fails to match either human perspective (III & IV) likely identify room for improvement.

4. QUALITATIVE ANALYSIS, IN FOUR PARTS

Using this suite of analysis tools described previously, a thorough exploration of the relationship between reference and estimated annotations is conducted, resulting in four significant insights. In the spirit of both reproducibility and open access, a companion IPython notebook is made available online⁹, providing additional visualizations complementary to the following discussion.

4.1 Invalid Harmonic Assumptions

An exploration of quadrant (IV) from Figure 1 reveals that a large source of error stems from musical content or reference chord annotations that violate basic assumptions about how chords are used. One common form of this behavior is due to issues of intonation, where a handful of recordings are not tuned to A440, with some varying by more than a quarter-tone: for example, “Stand By Me” by Jimmy Ruffin, “I’ll Tumble 4 Ya” by *The Culture Club*, “Every Breath You Take” by *The Police*, or “Nowhere to Run” by Martha Reeves and *the Vandellas*. Understandably, as a result, the estimated annotations differ by a semitone from the reference, and perform poorly across all comparison rules.

The second observation finds that some tracks in the dataset do not truly make use of, and are thus not well described by, chords. While a few classic songs by *The Beatles* have been known to be of questionable relevance for their instrumentation and lack of standard chords, such as “Revolution 9,” “Love You To,” or “Within You, Without You”, analysis here identifies several other tracks, spanning rap, hip hop, reggae, funk and disco, that behave similarly: for example, “Brass Monkey” by *The Beastie Boys*, “I, Me, & Myself” by *de la Soul*, “Don’t Push” by *Sublime*, “Get Up (I Feel Like Being a Sex Machine)” by James Brown, or “I Wanna Take You Higher” by Tina and Ike Turner. This realization encourages the conclusion that chords may not be a valid way to describe all kinds of music, and that using such songs for evaluation may lead to erroneous or misleading results.

4.2 Limitations of Chord Comparisons

The second observation resulting from this analysis is the difficulty faced in the comparison of related chords. By and large, ACE systems are often forced to either map chords to a finite dictionary, or develop embedding rules for equivalence testing [14]. In either case, this quantization process assigns all observations to a one-of- K representation effectively making all errors equivalent. For the purposes of stable evaluation, this can have significantly negative consequences.

⁹<https://github.com/ejhumphrey/ace-lessons/experiments.ipynb>

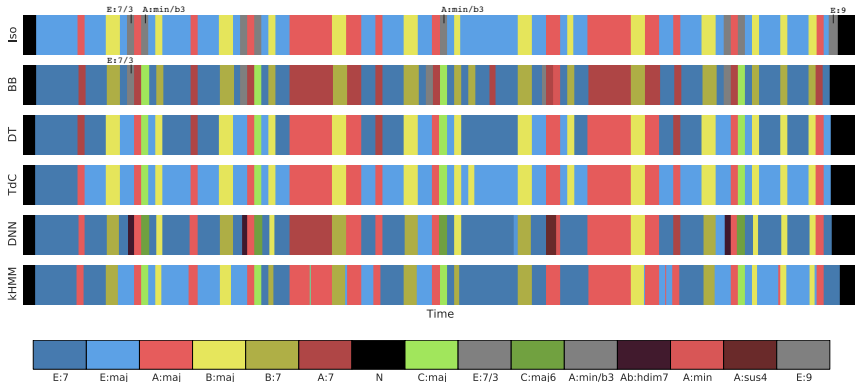


Figure 2. Six perspectives on “I Saw Her Standing There”, by *The Beatles*, according to Isophonics (Iso), Billboard (BB), David Temperley (DT), Trevor deClercq (TdC), the Deep Neural Network (DNN), and the k-stream HMM (kHMM).

Chords are naturally related to each other hierarchically, and cannot always be treated as distinct classes. Flat classification problems —i.e. those in which different classes are disjoint— are built on the assumption of mutually exclusive relationships. In other words, assignment to one class precludes the valid assignment to any other class considered. In the space of chords, $C:\dim7$ and $C:maj$ are perhaps mutually exclusive classes, but it is difficult to say the same of $C:maj7$ and $C:maj$, as the former *contains* the latter. This conflict is a common source of disagreement between annotators of the Rock Corpus tracks, which are easily identified in or near the quadrant (II) of Figure 1-b: for example, “Dancing In The Street” by Martha Reeves & The Vandellas, “All Apologies” by *Nirvana*, or “Papa’s Got a Brand New Bag” by James Brown. In each case, the human perspectives each report related tetrads and triads, e.g. $E:7$ and $E:maj$, causing low annotator agreement, while the machine estimation alternates between the two trying to represent both. These kinds of errors are not “confusions” in the classic sense, but a limitation of evaluation methods to reliably quantify this behavior, and of the model to represent this naturally structured output.

4.3 Conflicting Problem Definitions

Over the years, the automatic prediction of chord sequences from music audio has taken several names: estimation, recognition, identification, or transcription. The analysis here motivates the notion that this is not merely a matter of semantics, but actually a subtle distinction indicative of two slightly different problems being addressed. Chord *transcription* is an abstract task related to functional analysis, taking into consideration high-level concepts such as long term musical structure, repetition, segmentation or key. Chord *recognition*, on the other hand, is quite literal, and is closely related to polyphonic pitch

detection. Both interpretations are easily found in the collection of reference annotations, however, conflating these two tasks to some unknown degree.

Furthermore, the goal in transcription is to assign chord labels to regions, and is closer in principle to segmentation than classic approaches to chord estimation. One illustrative instance, “All Apologies” by *Nirvana*, is identified in quadrant (II) of Figure 1. Here, the human annotators have disagreed on the harmonic spelling of the entire verse, with DT and TdC reporting $C:maj$ and $C#:7$, respectively. On closer inspection, it would appear that both annotators are in some sense correct: the majority of the verse is arguably $C#:maj$, but a cello sustains the flat- 7^{th} of this key intermittently. The regions in which this occurs are clearly captured in the estimated annotations, corresponding to its $C#:7$ predictions. This proves to be an interesting discrepancy, because one annotator (DT) is using long-term structural information about the song to apply a single chord to the entire verse.

4.4 Ground Truth vs. Subjectivity

While the role that subjectivity can play in chord estimation is becoming better understood [12], it is not handled gracefully in current ACE methodology, and there are two examples worth analyzing here. The first, “I Saw Her Standing There” by *The Beatles*, is given in Figure 2, where the pitch class of the chord’s root is mapped to color hue, and the darkness is a function of chord quality, e.g., all $E:*$ chords are a shade of blue. No-chords are always black, and chords that do not fit into one of the 157 chord classes are shown in gray. Perhaps the most striking observation is the degree of variance between all annotations. Based on the tetrads comparison, no two reference annotations correspond to greater than a 65% agreement, with the DNN and kHMM scoring 28% and 52%

Ver.	Chord Sequence				Score	Ratings	Views
Billboard	D:maj	A:sus4 (b7)	B:min7	G:maj9	—	—	—
MARL	D:maj	D:maj/5	D:maj6/6	D:maj (4) /4	—	—	—
DT	D:maj	A:maj	B:min	G:maj	—	—	—
TdC	D:maj	A:maj	B:min	G:maj	—	—	—
DNN	D:maj	A:sus4	B:min7	G:maj7	—	—	—
kHMM	D:maj	A:sus4	B:min	G:maj	—	—	—
1	D:maj	A:maj	B:min	G:maj	4/5	193	1,985,878
2	D:5	A:sus4	B:min7	G:maj	5/5	11	184,611
3*	D:maj	A:maj	B:min	G:maj	4/5	23	188,152
4*	D:maj	A:maj	B:min	G:maj7	4/5	14	84,825
5*	D:maj	A:maj	B:min	G:maj	5/5	248	338,222
6	D:5	A:5	D:5/B	G:5	5/5	5	16,208

Table 3. Various interpretations of the verse from “With or Without You” by U2, comparing the reference annotations and automatic estimations with six interpretations from a popular guitar tablature website; a raised asterisk indicates the transcription is given relative to a capo, and transposed to the actual key here.

against the ground truth Isophonics reference, shown at the top. Despite this low score, the DNN and kHMM estimations agree with at least one of the four human annotations 89.1% and 92.3% of the song, respectively. The two exceptions occur during the out-of-gamut chords, E:7/3 and A:min/b3, which the DNN calls Ab:hdim7 and C:maj6, respectively. While both estimated chords share three pitches with the Isophonics reference, the other human annotators mark the A:min/b3 instead as a root position C:maj. Given how subjective it might be for human experts to agree on possible inversions, typical evaluation strategies may place too much emphasis on the root of a chord.

A second example to consider in the larger discussion of subjectivity is the verse of “With or Without You” by U2. Musically, one finds reasonably ambiguous harmonic content, consisting of a vocal melody, a moving bass line, a guitar riff, and a string pad sustaining a high-pitched D. Complementing the four expert perspectives provided here, an Internet search yields six additional user-generated chord transcriptions from the website Ultimate Guitar¹⁰. All human perspectives and both machine interpretations are consolidated in Table 3, noting both the average and number of ratings, as well as the number of views the public chord annotation has received. Though view count is not directly indicative of a transcription’s accuracy, it does provide a weak signal indicating that users did *not* rate it negatively.

This particular example provides several valuable insights. Nearly all perspectives are equivalent at the major-minor level, with the exception of the MARL annotation, which differs only slightly. That said, the differences between user-generated annotations do not noticeably impact the average ratings. This is an important consideration when building user systems, whereby objective measures are valuable insofar as they correlate with subjective experience. Similarly, these annotations are indicative of, at least for this song, a preference for root position chords. Thus, subjectivity plays a role in the collection of reference annotations, as well as the end-user experience.

¹⁰ http://tabs.ultimate-guitar.com/u/u2/with_or_without_you_crd.htm, accessed 19 April 2015.

5. CONCLUSIONS AND FUTURE PERSPECTIVES

In this work, qualitative analysis of system performance led to the identification of four key observations affecting current chord estimation methodology: one, not all music content is valid in the context of chord estimation; two, conventional comparison methods struggle to accurately characterize the complex relationships between chords; three, conventional methodology has mixed the somewhat conflicting goals of chord transcription and recognition to an undefined degree; and four, the subjective nature of chord perception may render objective ground truth and evaluation untenable.

Looking to the future of automatic chord estimation, a few opportunities stand out. First and foremost, subjectivity in reference annotations should be embraced rather than resolved. Chord estimation may be better understood as a time-aligned “tagging” problem, modeled as multinomial regression, or as structured prediction. Furthermore, synthesizing multiple human perspectives into a continuous-valued chord affinity vector would allow for more stable evaluation by encoding the degree to which a chord label applies to an observation. From a system design perspective, chord transcription, as a distinct task, stands to benefit greatly from recent advances in music structure analysis. To the point, however, it is also crucial to distinguish between the different flavors of harmonic analysis, and how a collection of reference data does—or does not—reflect the specific problem being addressed.

In a more general sense, this inquiry also has implications for the larger field of content-based MIR. Perhaps most pressing, the most powerful model cannot compensate for methodological deficiencies, and domain knowledge can be crucial to help understand system behaviour. Similarly, qualitative evaluation should play a larger role in the assessment of automatic systems intended for user-facing applications. If nothing else, users studies can help identify objective measures that align well with subjective experience. Finally, on a more practical note, high-performing systems can and should be used to facilitate the curation of reference annotations. These systems can be used to solicit human perspectives at a much larger scale, for both new and previously annotated content.

6. REFERENCES

- [1] Nicolas Boulanger-Lewandowski, Yoshua Bengio, and Pascal Vincent. Audio chord recognition with recurrent neural networks. In *Proceedings of the 14th International Society for Information Retrieval Conference (ISMIR)*, pages 335–340, 2013.
- [2] John Ashley Burgoyne, Jonathan Wild, and Ichiro Fujinaga. An expert ground truth set for audio chord recognition and music analysis. In *Proceedings of the 12th International Society for Information Retrieval Conference (ISMIR)*, pages 633–638, 2011.
- [3] Taemin Cho. *Improved techniques for automatic chord recognition from music audio signals*. PhD thesis, New York University, 2014.
- [4] Taemin Cho and Juan Pablo Bello. On the relative importance of individual components of chord recognition systems. *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, 22(2):477–492, 2014.
- [5] Taemin Cho, Ron J. Weiss, and Juan Pablo Bello. Exploring common variations in state of the art chord recognition systems. In *Proceedings of the Sound and Music Computing Conference*, 2010.
- [6] Trevor De Clercq and David Temperley. A corpus analysis of rock harmony. *Popular Music*, 30(01):47–70, 2011.
- [7] Takuya Fujishima. Realtime chord recognition of musical sound: a system using common lisp music. In *Proceedings of the International Computer Music Conference (ICMC)*, 1999.
- [8] Christopher Harte, Mark B Sandler, Samer A Abdallah, and Emilia Gómez. Symbolic representation of musical chords: A proposed syntax for text annotations. In *Proceedings of the 6th International Society for Information Retrieval Conference (ISMIR)*, volume 5, pages 66–71, 2005.
- [9] Eric J. Humphrey. *An Exploration of Deep Learning in Music Informatics*. PhD thesis, New York University, 2015.
- [10] Eric J. Humphrey and Juan Pablo Bello. Rethinking automatic chord recognition with convolutional neural networks. In *Proceedings of the International Conference on Machine Learning and Applications*, 2012.
- [11] Meinard Müller and Sebastian Ewert. Towards Timbre-Invariant Audio Features for Harmony-Based Music. *IEEE Transactions on Audio, Speech and Language Processing*, 18(3):649–662, March 2010.
- [12] Yizhao Ni, Matt McVicar, Raul Santos-Rodriguez, and Tjil De Bie. Understanding effects of subjectivity in measuring chord estimation accuracy. *Transactions on Audio, Speech, and Language Processing*, 21(12):2607–2615, 2013.
- [13] Johan Pauwels and Geoffroy Peeters. Evaluating automatically estimated chord sequences. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 749–753. IEEE, 2013.
- [14] Colin Raffel, Brian McFee, Eric J Humphrey, Justin Salamon, Oriol Nieto, Dawen Liang, and Daniel PW Ellis. mir_eval: A transparent implementation of common MIR metrics. In *Proceedings of the 15th International Society for Information Retrieval Conference*, 2014.
- [15] José R Zapata, André Holzapfel, Matthew EP Davies, João Lobato Oliveira, and Fabien Gouyon. Assigning a confidence threshold on automatic beat annotation in large datasets. In *Proceedings of the 13th International Society for Information Retrieval Conference*, pages 157–162, 2012.

IMPROVING MELODIC SIMILARITY IN INDIAN ART MUSIC USING CULTURE-SPECIFIC MELODIC CHARACTERISTICS

Sankalp Gulati[†], Joan Serra^{*} and Xavier Serra[†]

[†]Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain

^{*}Telefonica Research, Barcelona, Spain

sankalp.gulati@upf.edu, joan.serra@telefonica.com, xavier.serra@upf.edu

ABSTRACT

Detecting the occurrences of rāgs' characteristic melodic phrases from polyphonic audio recordings is a fundamental task for the analysis and retrieval of Indian art music. We propose an abstraction process and a complexity weighting scheme which improve melodic similarity by exploiting specific melodic characteristics in this music. In addition, we propose a tetrachord normalization to handle transposed phrase occurrences. The melodic abstraction is based on the partial transcription of the steady regions in the melody, followed by a duration truncation step. The proposed complexity weighting accounts for the differences in the melodic complexities of the phrases, a crucial aspect known to distinguish phrases in Carnatic music. For evaluation we use over 5 hours of audio data comprising 625 annotated melodic phrases belonging to 10 different phrase categories. Results show that the proposed melodic abstraction and complexity weighting schemes significantly improve the phrase detection accuracy, and that tetrachord normalization is a successful strategy for dealing with transposed phrase occurrences in Carnatic music. In the future, it would be worthwhile to explore the applicability of the proposed approach to other melody dominant music traditions such as Flamenco, Beijing opera and Turkish Makam music.

1. INTRODUCTION

The automatic assessment of melodic similarity is one of the most researched topics in music information research (MIR) [3, 14, 30]. Melodic similarity models may vary considerably depending on the type of music material (sheet music or polyphonic audio recordings) [4, 8, 22] and the music tradition [5, 18]. Results until now indicate that the important characteristics of several melody-dominant music traditions of the world such as Flamenco and Indian art music (IAM) need dedicated research efforts to devise specific approaches for computing melodic similarity [23, 24]. These music traditions have large audio music repertoires but comparatively very fewer number of descriptive scores

(they follow an oral transmission), the automatic detection of the occurrences of a melodic phrase in audio recordings is therefore a task of primary importance. In this article, we focus on this task for IAM.

Hindustani music (also referred to as north Indian art music) and Carnatic music (also referred to as south Indian art music) are the two art music traditions of India [6, 31]. Both are heterophonic in nature, with melody as the dominant aspect of the music. A typical piece has a main melody being sung or played by the lead artist and a melodic accompaniment with the tonic pitch as the base reference frequency [9]. *Rāg* is the melodic framework and *tāl* is the rhythm framework in both music traditions. Rāgs are characterized by their constituent *svars* (roughly speaking, notes), by the *āroh-avroh* (the ascending and descending melodic progression) and, most importantly, by a set of characteristic melodic or 'catch' phrases. These phrases are the prominent cues for *rāg* identification used by the performer, to establish the identity of a *rāg*, and also the listener, to recognize the *rāg*.

The characteristic melodic phrases of a *rāg* act as the basis for the artists to improvise, providing them with a medium to express creativity during a *rāg* rendition. Hence, the surface representation of these melodic phrases can vary a lot across their occurrences. This high degree of variability in terms of the duration of a phrase, non-linear time warpings and the added melodic ornaments together pose a big challenge for melodic similarity computation in IAM. In Figure 1 we illustrate this variability by showing the pitch contours of the different occurrences of three characteristic melodic phrases of the *rāg* Alaiya Bilawal. We can clearly see that the duration of a phrase across its occurrences varies a lot and the steady melodic regions are highly varied in terms of the duration and the presence of melodic ornaments. Because of these and other factors, detecting the occurrences of characteristic melodic phrases becomes a challenging task. Ideally, the melodic similarity measure should be robust to a high degree of variation and, at the same time, it should be able to discriminate between different phrase categories and irrelevant melodic fragments (noise candidates).

For melodic similarity computation, the string matching-based and the set point-based approaches are extensively used for both musical scores and audio recordings [30]. However, compared to the former, the set point-based approaches are yet to be fully exploited for polyphonic audio music because of the challenges involved in melody extrac-



© Sankalp Gulati[†], Joan Serra^{*} and Xavier Serra[†]. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Sankalp Gulati[†], Joan Serra^{*} and Xavier Serra[†]. "Improving Melodic Similarity in Indian Art Music Using Culture-specific Melodic Characteristics", 16th International Society for Music Information Retrieval Conference, 2015.

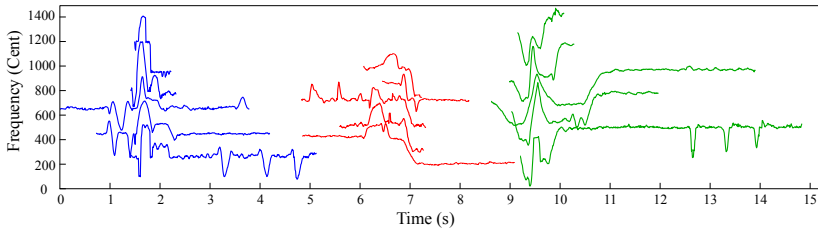


Figure 1. Pitch contours of occurrences of three different characteristic melodic phrases in Hindustani music. Contours are frequency transposed and time shifted for a better visualization.

tion and transcription [4]. A reliable melody transcription algorithm is argued to be the key to bridge the gap between audio and symbolic music, leading to the full exploitation of the potential of the set point-based approaches for audio music. However, for several music traditions such as Hindustani and Carnatic music, automatic melody transcription is a challenging and a rather ill-defined task [25].

In recent years, several methods for retrieving different types of melodic phrases have been proposed for IAM, following both supervised and unsupervised strategies [7, 12, 13, 16, 17, 24, 26, 27]. Ross et al. [27] detect the occurrences of the title phrases of a composition within a concert recording of Hindustani music. The authors explored a SAX-based representation [20] along with several pitch quantizations of the melody and showed that a dissimilarity measure based on dynamic time warping (DTW) is preferred over the Euclidean distance. Noticeably, in that work, the underlying rhythm structure was exploited to reduce the search space for detecting pattern occurrences. An extension of that approach [26] pruned the search space by employing a melodic landmark called *nyās svar* [11].

Rao et al. [24] address the challenge of a large within-class variability in the occurrences of the characteristic phrases. They propose to use exemplar-based matching after vector quantization-based training to obtain multiple templates for a given phrase category. In addition, the authors propose to learn the optimal DTW constraints in a previous step for each phrase category in order to exploit the possible patterns in the duration variability. For Carnatic music, Ishwar et al. [17] propose a two-stage approach for spotting the characteristic melodic phrases. The authors exploit specific melodic characteristics (saddle points) to reduce the target search space and use a distance measure based on rough longest common subsequence [19].

On the other hand, there are studies that follow an unsupervised approach for discovering melodic patterns in Carnatic music [7, 12]. Since the evaluation of melodic similarity measures is a much more challenging task in an unsupervised framework, results obtained from an exhaustive grid search of optimal distance measures and parameter values within a supervised framework are valuable [13].

In this study, we present two approaches that utilize specific melodic characteristics in IAM to improve melodic similarity. We propose a melodic abstraction process based

on the partial transcription of the melodies to handle large timing variations in the occurrences of a melodic phrase. For Carnatic music we also propose a complexity weighting scheme that accounts for the differences in the melodic complexities of the phrases, a crucial aspect for melodic similarity in this music tradition. In addition, we come up with a tetrachord normalization strategy to handle the transposed occurrences of the phrases. The dataset used for the evaluation is a superset of the dataset used in a recent study [13] and contains nearly 30% more number of annotated phrases.

2. METHOD

Before we present our approach we first discuss the motivation and rationale behind it. A close examination of the occurrences of the characteristic melodic phrases in our dataset reveals that there is a pattern in the non-linear timing variations [24]. In Figure 1 we show a few occurrences of three such melodic phrases. In particular, we see that the transient regions of a melodic phrase tend to span nearly the same time duration across different occurrences, whereas the stationary regions vary a lot in terms of the duration. In Figure 2 we further illustrate this by showing two occurrences of a melodic phrase (P_{1a} and P_{2a}). The stationary *svar* regions are highlighted. We clearly see that the duration variation is prominent in the highlighted regions. To handle such large non-linear timing variations typically a non-constrained DTW distance measure is employed [13]. However, such a DTW variant is prone to noisy matches. Moreover, the absence of a band constraint renders it inefficient for computationally complex tasks such as motif discovery [12].

We put forward an approach that abstracts the melodic representation and reduces the extent of duration and pitch variations across the occurrences of a melodic phrase. Our approach is based on the partial transcription of the melodies. As mentioned earlier, melodic transcription in IAM is a challenging task. The main challenges arise due to the presence of non-discrete pitch movements such as smooth glides and *gamakas*¹. However, since the duration variation exists mainly during the steady *svar* regions, transcribing only the stable melodic regions might be suffi-

¹ Rapid oscillatory melodic movement around a *svar*

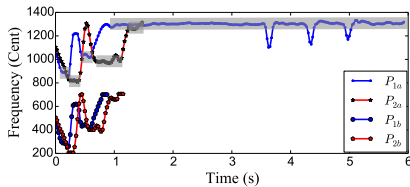


Figure 2. Original pitch contours (P_{1a} , P_{2a}) and duration truncated pitch contours (P_{1b} , P_{2b}) of two occurrences of a characteristic phrase of rāg Alhaiya Bilawal. The contours are transposed for a good visualization.

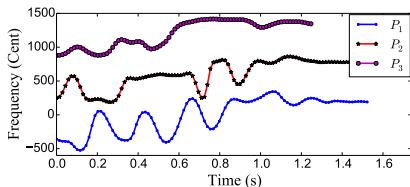


Figure 3. Pitch contours of three melodic phrases (p_1 , p_2 , p_3). p_1 and p_2 are the occurrences of the same characteristic phrase and both are musically dissimilar to p_3 .

cient. Once transcribed, we can then truncate the duration of these steady melodic regions and hence effectively reduce the amount of timing variations across the occurrences of a melodic phrase. Additionally, since the duration truncation also reduces the overall length of a pattern, the computational time for melodic similarity computation is also reduced substantially.

The rapid oscillatory pitch movements (gamakas) in Carnatic music bring up another set of challenges for the melodic similarity computation. Very often, two musically dissimilar melodic phrases obtain a high similarity score owing to a similar pitch contour at a macro level. However, they differ significantly at a micro level. In Figure 3 we illustrate such a case where we show the pitch contours of three melodic phrases P_1 , P_2 and P_3 , where P_1 and P_2 are the occurrences of the same melodic phrase and both are musically dissimilar to P_3 . Using the best performing variant of the similarity measure in [13] we obtain a higher similarity score between the pairs (P_1 , P_3) and (P_2 , P_3) compared to the score between the pair (P_1 , P_2). This tendency of a high complexity time-series (higher degree of micro level variations) obtaining a high similarity score with another low complexity time-series is discussed in [1]. We follow their approach and apply a complexity weighting to account for the differences in the melodic complexities between phrases in the computation of melodic similarity.

In the subsequent sections we present our proposed approach. As a baseline in this study we consider the method that was reported as the best performing method in a recent study for the same task on a subset of the dataset [13]. We

denote this baseline method by M_B .

2.1 Melody Estimation and post-processing

We represent melody of an audio signal by the pitch of the predominant melodic source. We use the method proposed by Salamon and Gómez [29]. This method performed favourably in MIREX 2011 (an international MIR evaluation campaign) on a variety of music genres, including IAM, and has been used in several other studies for a similar task [7, 12, 13]. An implementation of this algorithm available in Essentia [2] is used in this study. Essentia is an open-source C++ library for audio analysis and content-based MIR. We use the default values of the parameters for pitch estimation except the frame size and the hop size, which are set to 46 ms and 2.9 ms, respectively. For Hindustani music, we use the pitch tracks corresponding to the predominant melody that are used in several other studies on a similar topic [24, 27] and are made available to us by the authors. These pitch tracks are obtained using a semi-automatic system for predominant melody estimation. This allows us to compare results across studies and avoid the effects of pitch errors on the computation of melodic similarity. After estimating the predominant pitch we convert it from Hertz to Cent scale for the melody representation to be musically relevant.

We proceed to post-process the pitch contours and remove the spurious pitch jumps lasting over a few frames as well as smooth the pitch contours. We first apply a median filter over a window size of 50 ms, followed by a low-pass filter using a Gaussian window. The window size and the standard deviation of the Gaussian window is set to 50 ms and 10 ms, respectively. The pitch contours are finally down-sampled to 100 Hz, which was found to be an optimal sampling rate in [13].

2.2 Transposition Invariance

The base frequency chosen for a melody in IAM is the tonic pitch of the lead artist [10]. Therefore, for a meaningful comparison of the melodic phrases across the recordings of different artists, a melody representation should be normalized by the tonic pitch of the lead artist. We perform this tonic normalization (N_{tonic}) by considering the tonic of the lead artist as the reference frequency during the Hertz to Cent conversion. The tonic pitch is automatically identified using a multi-pitch approach proposed by Gulati et al. [10]. This approach was shown to obtain more than 90% tonic identification accuracy and has been used in several studies in the past.

Tonic normalization does not account for the pitch of the octave transposed occurrences of a melodic phrase within a recording. In addition, estimated tonic pitch sometimes might be incorrect and a typical error is an offset of fifth scale degree. To handle such cases, we propose a novel tetrachord normalization (N_{tetra}). For this we analyse the difference (Δ) in the mean frequency values of the two tonic normalized melodic phrases (p_1 , p_2). We offset the pitch values of the phrase p_1 by the frequency in the set $\{-1200, -700, -500, 0, 500, 700, 1200, 1700, 1900\}$ that

is closest to Δ within a vicinity of 100 Cents. In addition to tetrachord normalization, we also experiment with mean normalization (N_{mean}), which was reported to improve the performance in the case of Carnatic music [13].

2.3 Partial Transcription

We perform a partial melody transcription to automatically segment and identify the steady svar regions in the melody. Note that even the partial transcription of the melodies is a non-trivial task, since we desire a segmentation that is robust to different melodic ornaments added to a svar where the pitch deviation from the mean svar frequency can be up to 200 Cents. In Figure 2 we show such an example of a steady svar region (P_{1a} from 3-6 s) where the pitch deviation from the mean svar frequency is high due to added melodic ornaments. Ideally, the melodic region between 1 and 6 s should be detected as a single svar segment.

We segment the steady svar regions using a method described in [11], which addresses the aforementioned challenges. A segmented svar region is then assigned a frequency value corresponding to the peak in an aggregated pitch histogram closest to the mean svar frequency. The pitch histogram is constructed for the entire recording and smoothed using a Gaussian window with a variance of 15 cents. As peaks of the normalized pitch histogram, we select all the local maximas where at least one peak-to-valley ratio is greater than 0.01. For a detailed description of this method we refer to [11].

2.4 Svar Duration Truncation

After segmenting the steady svar regions in the melody we proceed to truncate the duration of these regions. We hypothesize that, beyond a certain value δ , the duration of these steady svar regions do not change the identity of a melodic phrase (i.e. the phrase category). We experiment with 7 different truncation durations $\delta = \{0.1 \text{ s}, 0.3 \text{ s}, 0.5 \text{ s}, 0.75 \text{ s}, 1 \text{ s}, 1.5 \text{ s}, 2 \text{ s}\}$ and select the one that results in the best performance. In Figure 2 we show an example of the occurrences of a melodic phrase both before (P_{1a} , P_{2a}) and after (P_{1b} , P_{2b}) the svar duration truncation using $\delta = 0.1 \text{ s}$. This example clearly illustrates that the occurrences of a melodic phrase after duration truncation exhibit lower degree of non-linear timing variations. We denote this method by M_{DT} .

2.5 Similarity Computation

To measure the similarity between two melodic fragments we consider a DTW-based approach. Since the phrase segmentation is known beforehand, we use a whole sequence matching DTW variant. We consider the best performing DTW variant and the related parameter values for each music tradition as reported in [13]. These variants were chosen based on an exhaustive grid search across all possible combinations and hence can be considered as optimal for this dataset. For Carnatic music we use a DTW step size condition $\{(2, 1), (1, 1), (1, 2)\}$ and for Hindustani music a step size condition $\{(1, 0), (1, 1), (0, 1)\}$. We use Sakoe-Chiba global band constraint [28] with the width of the

Dataset	Rec.	PC	Rāgs	Artists	Duration (hr)
CMD	23	5	5	14	3.82
HMD	9	5	1	7	1.76

Table 1. Details of the datasets in terms of the total number of recordings (Rec.), number of annotated phrase categories (PC), number of rāgs, unique number of artists and total duration of the dataset.

band as $\pm 10\%$ of the phrase length. Note that before computing the DTW distance we uniformly time-scale the two melodic fragments to the same length, which is the maximum of the lengths of the phrases.

2.6 Complexity Weighting

The complexity weighting that we apply here to overcome the shortcoming of the distance measure in distinguishing two time series with different complexities is discussed in [1]. We apply a complexity weighting (α) to the DTW-based distance (D_{DTW}) in order to compute the final similarity score $D_f = \alpha D_{DTW}$. We compute α as:

$$\alpha = \frac{\max(C_i, C_j)}{\min(C_i, C_j)}; \quad C_i = \sqrt{\sum_{i=1}^{N-1} (p_i - p_{i+1})^2} \quad (1)$$

where, C_i is the complexity estimate of a melodic phrase of length N samples and p_i is the pitch value of the i^{th} sample. We explore two variants of this complexity estimate. One of these variants is already proposed in [1] and is described in equation 1. We denote this method variant by M_{CW1} . We propose another variant that utilizes melodic characteristics of Carnatic music. This variant takes the number of saddle points in the melodic phrase as the complexity estimate [17]. This method variant is denoted by M_{CW2} . As saddle points we consider all the local minimas and the local maximas in the pitch contour which have at least one minima to maxima distance of half a semitone. Since such melodic characteristics are predominantly present in Carnatic music, the complexity weighting is not applicable for computing melodic similarity in Hindustani music.

3. EVALUATION

3.1 Dataset and Annotations

For a better comparison of the results, for our evaluations we use a music collection that has been used in several other studies for a similar task [13, 24, 27]. However, we have extended the dataset by adding 30% more number of annotations of the melodic phrases, which we make available at <http://compmusic.upf.edu/node/269>. The music collection comprises vocal recordings of renowned artists in both Hindustani and Carnatic music. We use two separate datasets for the evaluation, Carnatic music dataset (CMD) and Hindustani music dataset (HMD) as done in [13]. The melodic phrases are annotated by two professional musicians who have received over 15 years of formal music training. All the annotated phrases are the characteristic

CMD				HMD			
PC	#Occ	L_{mean}	L_{std}	PC	#Occ	L_{mean}	L_{std}
C_1	39	1.38	0.25	H_1	62	1.93	0.98
C_2	46	1.25	0.21	H_2	154	1.40	0.79
C_3	38	1.23	0.24	H_3	47	1.30	0.78
C_4	31	1.11	0.17	H_4	76	2.38	1.33
C_5	45	0.76	0.08	H_5	87	1.17	0.36
Total	199	1.13	0.29		426	1.59	0.99

Table 2. Details of the 625 annotated melodic phrases. PC: pattern category, #Occ: number of annotated occurrences, and L_{mean} , L_{std} are the mean, standard deviation of the lengths of the patterns of a PC in seconds.

phrases of a rāg. In Table 1 we summarize the relevant dataset details. Table 2 summarizes the details of the annotated phrases in terms of their number of instances and basic statistics of the length of the phrases.

3.2 Setup, Measures and Statistical Significance

We consider each annotated melodic phrase as a query and perform a search across all the annotated phrases in the dataset (referred to as target search space). In addition to the annotated phrases, we add randomly sampled melodic segments (referred to as noise candidates) in the target space to simulate a real world scenario. We generate the starting time stamps of the noise candidates by randomly sampling a uniform distribution. The length of the noise candidates are generated by sampling the distribution of the duration values of the annotated phrases. The number of noise candidates added are 100 times the number of total annotations in the entire music collection. For every query we consider the top 1000 nearest neighbours in the search results ordered by the similarity value. A retrieved melodic phrase is considered as a true hit only if it belongs to the same phrase category as the query.

To assess the performance of the proposed approach and the baseline method we use mean average precision (MAP), a common measure in information retrieval [21]. To assess if the difference in the performance of any two methods is statistically significant we use the Wilcoxon signed rank-test [32] with $p < 0.01$. To compensate for multiple comparisons, we apply the Holm-Bonferroni method [15].

4. RESULTS AND DISCUSSION

In Table 3 we summarize the MAP scores and the standard deviation of the average precision values obtained using the baseline method (M_B), the method that uses duration truncation (M_{DT}) and the ones using the complexity weighting (M_{CW1} , M_{CW2}), for both the CMD and the HMD. Note that M_{CW1} and M_{CW2} are only applicable to the CMD (Sec. 2).

We first analyse the results for the HMD. From Table 3 (upper half), we see that the proposed method variant that applies a duration truncation performs better than the baseline method for all the normalization techniques. More-

HMD				
Norm	M_B	M_{DT}	M_{CW1}	M_{CW2}
N_{tonic}	0.45 (0.25)	0.52 (0.24)	-	-
N_{mean}	0.25 (0.20)	0.31 (0.23)	-	-
N_{tetra}	0.40 (0.23)	0.47 (0.23)	-	-
CMD				
Norm	M_B	M_{DT}	M_{CW1}	M_{CW2}
N_{tonic}	0.39 (0.29)	0.42 (0.29)	0.41 (0.28)	0.41 (0.29)
N_{mean}	0.39 (0.26)	0.45 (0.28)	0.43 (0.27)	0.45 (0.27)
N_{tetra}	0.45 (0.26)	0.50 (0.27)	0.49 (0.28)	0.51 (0.27)

Table 3. MAP scores for the two datasets HMD and CMD for the four method variants M_B , M_{DT} , M_{CW1} and M_{CW2} and for different normalization techniques. Standard deviation of average precision is reported within round brackets.

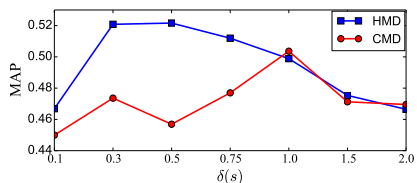


Figure 4. MAP scores for different duration truncation values (δ) for the HMD and the CMD.

over, this difference is found to be statistically significant in each case. The results for the HMD in this table correspond to $\delta = 500$ ms, for which we obtain the highest accuracy compared to the other δ values as shown in Figure 4. Furthermore, we see that N_{tonic} results in the best accuracy for the HMD for all the method variants and the difference is found to be statistically significant in each case. In Figure 5 we show a boxplot of average precision values for each phrase category and for both M_B and M_{DT} to get a better understanding of the results. We observe that with an exception of the phrase category H_2 , M_{DT} consistently performs better than M_B for all the other phrase categories. A close examination of this exception reveals that the error often is in the segmentation of the steady svar regions of the melodic phrases corresponding to H_2 . This can be attributed to a specific subtle melodic movement in H_2 that is confused by the segmentation method as a melodic ornament instead of a svar transition, leading to a segmentation error.

We now analyse the results for the CMD. From Table 3 (lower half), we see that using the method variants M_{DT} , M_{CW1} and M_{CW2} we obtain reasonably higher MAP scores compared to the baseline method M_B and the difference is found to be statistically significant for each method variant across all normalization techniques. This MAP score for M_{DT} corresponds to $\delta = 1$ s, which is considerably higher than the MAP scores for other δ values as shown in Figure 4. We also see that M_{CW2} performs slightly better than M_{CW1} and the difference is found to

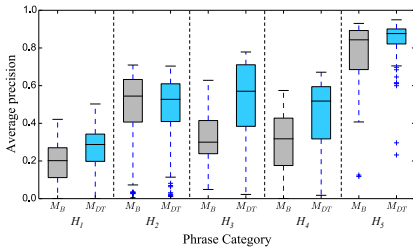


Figure 5. Boxplot of average precision values obtained using M_B and M_{DT} for each melodic phrase category for the HMD. These values correspond to N_{tonic} .

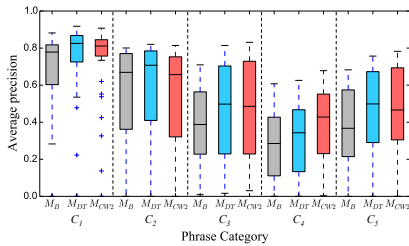


Figure 6. Boxplot of average precision values obtained using methods M_B , M_{DT} and M_{CW} for each melodic phrase category for the CMD. These values correspond to N_{tetra} .

be statistically significant only in the case of N_{tetra} . We do not find any statistically significant difference in the performance of methods M_{DT} and M_{CW2} . Unlike the HMD, for the CMD N_{tetra} results in the best performance with a statistically significant difference compared to the other normalization techniques across all method variants. We now analyse the average precision values for every phrase category for M_B , M_{DT} and M_{CW2} . Since M_{CW2} performs slightly better than M_{CW1} we only consider M_{CW2} for this analysis. In Figure 6 we see that M_{DT} performs better than M_B for all phrase categories. We also observe that M_{CW2} consistently performs better than M_B with the sole exception of C_2 . This exception occurs because M_{CW2} presumes a consistency in terms of the number of saddle points across the occurrences of a melodic phrase, which does not hold true for C_2 . This is because phrases corresponding to C_2 are rendered very fast and the subtle pitch movements are not the characteristic aspect of such melodic phrases. Hence, the artists often take the liberty of changing the number of saddle points.

Overall we see that duration truncation of steady melodic regions improves the performance in both the HMD and the CMD. This reinforces our hypothesis that elongation of steady svar regions in the melodies of IAM in the context of the characteristic melodic phrase does not change the musical identity of the phrase. This correlates

with the concept of nyās svar (nyās literally means home), where the artist has the flexibility to stay and elongate a single svar. A similar observation was reported in [24], where the authors proposed to learn the optimal global DTW constraints a priori for each pattern category. However, their proposed solution could not improve the performance. Further comparing the results for the HMD and the CMD we notice that N_{tonic} results in the best performance for the HMD and N_{tetra} for the CMD. This can be attributed to the fact that the number of the pitch-transposed occurrences of a melodic phrase is significantly higher in the CMD compared to the HMD [13]. Also, since the non-linear timing variability in the HMD is very high, any normalization (N_{mean} or N_{tetra}) that involves a decision based on the mean frequency of the phrase is more likely to fail.

5. CONCLUSIONS

In this paper we briefly presented an overview of the approaches for detecting the occurrences of the characteristic melodic phrases in audio recordings of Indian art music. We highlighted the major challenges involved in this task and focused on two specific issues that arise due to large non-linear timing variations and rapid melodic movements. We proposed simple and easy to implement solutions based on partial transcription and complexity weighting to address these challenges. We also put forward a new dataset by appending 30% more number of melodic phrase annotations to those used in previous studies. We showed that duration truncation of the steady svar regions in the melodic phrases results in a statistically significant improvement in the computation of melodic similarity. This confirms our hypothesis that the elongation of steady svar regions beyond a certain duration does not affect the perception of the melodic similarity in the context of the characteristic melodic phrases. Furthermore, we showed that complexity weighting significantly improves the melodic similarity in Carnatic music. This suggests that the extent and the number of saddle points is an important characteristic of a melodic phrase and is crucial to melodic similarity in Carnatic music.

In the future, we plan to improve the method used for segmenting the steady svar regions so that it can differentiate melodic ornaments from subtle svar transitions. In addition, we see a vast scope in further refining the complexity estimate of a melodic phrase to improve the complexity weighting. It would also be worthwhile to explore the applicability of this approach to music traditions such as Flamenco, Beijing opera and Turkish Makam music.

6. ACKNOWLEDGMENTS

This work is partly supported by the European Research Council under the European Unions Seventh Framework Program, as part of the CompMusic project (ERC grant agreement 267583). We thank Kaustav K. Ganguli and Vignesh Ishwar for the annotations and valuable discussions and, Ajay Srinivasamurthy for the proof-reading.

7. REFERENCES

- [1] G. E. Batista, X. Wang, and E. J. Keogh. A complexity-invariant distance measure for time series. In *SDM*, volume 11, pages 699–710, 2011.
- [2] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra. Essentia: an audio analysis library for music information retrieval. In *Proc. of Int. Society for Music Information Retrieval Conf. (ISMIR)*, pages 493–498, 2013.
- [3] M. A. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney. Content-based music information retrieval: Current directions and future challenges. *Proc. of the IEEE*, 96(4):668–696, 2008.
- [4] T. Collins, S. Böck, F. Krebs, and G. Widmer. Bridging the audio-symbolic gap: The discovery of repeated note content directly from polyphonic music audio. In *Audio Engineering Society's 53rd Int. Conf. on Semantic Audio*, 2014.
- [5] D. Conklin and C. Anagnostopoulou. Comparative Pattern Analysis of Cretan Folk Songs. *Journal of New Music Research*, 40(2):119–125, 2010.
- [6] A. Danielou. *The ragas of Northern Indian music*. Munshiram Manoharlal Publishers, New Delhi, 2010.
- [7] S. Dutta and H. A. Murthy. Discovering typical motifs of a raga from one-liners of songs in Carnatic music. In *Int. Society for Music Information Retrieval*, pages 397–402, 2014.
- [8] A. Ghias, J. Logan, D. Chamberlin, and B. C. Smith. Query by humming: musical information retrieval in an audio database. In *Proc. of the third ACM Int. Conf. on Multimedia*, pages 231–236. ACM, 1995.
- [9] S. Gulati. A tonic identification approach for Indian art music. Master's thesis, Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain, 2012.
- [10] S. Gulati, A. Bellur, J. Salamon, H. G. Ranjani, V. Ishwar, H. A. Murthy, and X. Serra. Automatic tonic identification in Indian art music: approaches and evaluation. *Journal of New Music Research*, 43(1):55–73, 2014.
- [11] S. Gulati, J. Serrà, K. K. Ganguli, and X. Serra. Landmark detection in hindustani music melodies. In *Int. Computer Music Conf., Sound and Music Computing Conf.*, pages 1062–1068, 2014.
- [12] S. Gulati, J. Serrà, V. Ishwar, and X. Serra. Mining melodic patterns in large audio collections of indian art music. In *Int. Conf. on Signal Image Technology & Internet Based Systems (SITIS-MIRA)*, pages 264–271, 2014.
- [13] S. Gulati, J. Serrà, and X. Serra. An evaluation of methodologies for melodic similarity in audio recordings of indian art music. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, pages 678–682, 2015.
- [14] W. B. Hewlett and E. Selfridge-Field. *Melodic similarity: Concepts, procedures, and applications*, volume 11. The MIT Press, 1998.
- [15] S. Holm. A simple sequentially rejective multiple test procedure. *Scandinavian journal of statistics*, 6(2):65–70, 1979.
- [16] V. Ishwar, A. Bellur, and H. A. Murthy. Motivic analysis and its relevance to raga identification in carnatic music. In *Proceedings of the 2nd CompMusic Workshop*, pages 153–157, 2012.
- [17] V. Ishwar, S. Dutta, A. Bellur, and H. Murthy. Motif spotting in an Alapana in Carnatic music. In *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, pages 499–504, 2013.
- [18] Z. Juhász. Motive identification in 22 folksong corpora using dynamic time warping and self organizing maps. In *Int. Society for Music Information Retrieval Conf.*, pages 171–176, 2009.
- [19] H. J. Lin, H. H. Wu, and C. W. Wang. Music matching based on rough longest common subsequence. *J. Inf. Sci. Eng.*, 27(1):95–110, 2011.
- [20] Jessica Lin, Eamonn Keogh, Stefano Lonardi, and Bill Chiu. A symbolic representation of time series, with implications for streaming algorithms. In *Proc. of the 8th ACM SIGMOD workshop on Research issues in data mining and knowledge discovery*, pages 2–11, 2003.
- [21] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to information retrieval*, volume 1. Cambridge university press Cambridge, 2008.
- [22] A. Marsden. Interrogating Melodic Similarity: A Definitive Phenomenon or the Product of Interpretation? *Journal of New Music Research*, 41(4):323–335, 2012.
- [23] A. Pikrakis, J. Mora, F. Escobar, and S. Oramas. Tracking melodic patterns in Flamenco singing by analyzing polyphonic music recordings. In *Proc. of Int. Society for Music Information Retrieval Conf. (ISMIR)*, pages 421–426, 2012.
- [24] P. Rao, J. C. Ross, K. K. Ganguli, V. Pandit, V. Ishwar, A. Bellur, and H. A. Murthy. Classification of melodic motifs in raga music with time-series matching. *Journal of New Music Research*, 43(1):115–131, 2014.
- [25] S. Rao. Culture Specific Music Information Processing : A Perspective From Hindustani Music. In *2nd Comp-Music Workshop*, pages 5–11, 2012.
- [26] J. C. Ross and P. Rao. Detection of raga-characteristic phrases from Hindustani classical music audio. In *Proc. of 2nd CompMusic Workshop*, pages 133–138, 2012.
- [27] J. C. Ross, T. P. Vinutha, and P. Rao. Detecting melodic motifs from audio for Hindustani classical music. In *Proc. of Int. Conf. on Music Information Retrieval (ISMIR)*, pages 193–198, 2012.
- [28] H. Sakoe and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. on Acoustics, Speech, and Language Processing*, 26(1):43–50, 1978.
- [29] J. Salamon and E. Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(6):1759–1770, 2012.
- [30] Rainer Typke. *Music retrieval based on melodic similarity*. 2007.
- [31] T. Viswanathan and M. H. Allen. *Music in South India*. Oxford University Press, 2004.
- [32] F. Wilcoxon. Individual comparisons by ranking methods. *Biometrics bulletin*, pages 80–83, 1945.

SEARCHING LYRICAL PHRASES IN A-CAPELLA TURKISH MAKAM RECORDINGS

Georgi Dzhabazov, Sertan Şentürk, Xavier Serra

Music Technology Group, Universitat Pompeu Fabra, Barcelona

{georgi.dzhabazov, sertan.senturk, xavier.serra}@upf.edu

ABSTRACT

Search by lyrics, the problem of locating the exact occurrences of a phrase from lyrics in musical audio, is a recently emerging research topic. Unlike key-phrases in speech, lyrical key-phrases have durations that bear important relation to other musical aspects like the structure of a composition. In this work we propose an approach that address the differences of syllable durations, specific for singing. First a phrase is expanded to MFCC-based phoneme models, trained on speech. Then, we apply dynamic time warping between the phrase and audio to estimate candidate audio segments in the given audio recording. Next, the retrieved audio segments are ranked by means of a novel score-informed hidden Markov model, in which durations of the syllables within a phrase are explicitly modeled. The proposed approach is evaluated on 12 a-capella audio recordings of Turkish Makam music. Relying on standard speech phonetic models, we arrive at promising results that outperform a baseline approach unaware of lyrics durations. To the best of our knowledge, this is the first work tackling the problem of search by lyrical key-phrases. We expect that it can serve as a baseline for further research on singing material with similar musical characteristics.

1. INTRODUCTION

Searching by lyrics is the problem of locating the exact occurrences of a key-phrase from textual lyrics in musical signal. It has inherent relation to the equivalent problem of keyword spotting (KWS) in speech. In KWS, a user is interested to find at which time position a relevant keyword (presenting a topic of interest) is spoken [16].

Most of the work on searching for keywords/key-phrases in singing (a.k.a lyrics spotting) has borrowed concepts from KWS. For spoken utterances phonemes have relatively similar duration across speakers. Unlike that, in singing durations of phonemes (especially vowels) have higher variation [8]. When being sung, vowels are prolonged according to musical note values. Therefore, adopt-

ing an approach from speech recognition might lack some singing-specific semantics, among which the durations of sung syllables. Furthermore, key-phrase detection has high potential to be integrated with other relevant MIR-applications, because lyrical key-phrases are often correlated to musical structure: For most types of music a section-long lyrical phrase is a feature that represents the corresponding structural section (e.g. chorus) in a unique way. Therefore correctly retrieved audio segments for, for example, the first lyrics line for a chorus can serve as a structure discovery tool.


In this work we investigate searching by lyrics in the case when a query represents an entire section or phrase from the textual lyrics of a particular composition. Unlike most works on lyrics spotting or query-by-humming, where a hit would be a document from an entire collection, in our case a hit is the occurrence of a phrase, being retrieved only from all performances of the given composition. In this respect the problem setting is more similar to linking melodic patterns from score to musical audio (addressed in [15]), rather than to lyrics spotting. We assume that the musical score with lyrics is present for the composition of interest. The proposed approach has been tested on a small dataset of a-cappella performances from a repertoire of Turkish Makam music. For a given performance, the composition is known in advance, but no information about the structure is given. Characteristic for Makam music is that, in a performance there might be reordering or repetitions of score sections.

2. RELATED WORK

2.1 Lyrics spotting

A recent work proved that lyrics spotting is a hard problem even when singing material is a-capella (for pop songs in English) [8]. The authors adopt an approach from KWS, using a compound hidden Markov model (HMM) with keyword and filler model. Keywords are automatically extracted from a textual collection of lyrics. This work's best classifier (multi-layer perceptron) yielded an f-measure of 44%, averaged over top 50% of keywords. Notably, the achieved results on singing material are not very different from results on spoken utterances of same keywords.

One of the few attempts to go beyond keywords is the work of [4]. Their goal was to automatically link phrases that appear in the lyrics of one song to the same phrase in another song. To this end, a keyword-filler model is

 © Georgi Dzhabazov, Sertan Şentürk, Xavier Serra . Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Georgi Dzhabazov, Sertan Şentürk, Xavier Serra. "Searching Lyrical Phrases in A-capella Turkish Makam Recordings". 16th International Society for Music Information Retrieval Conference, 2015.

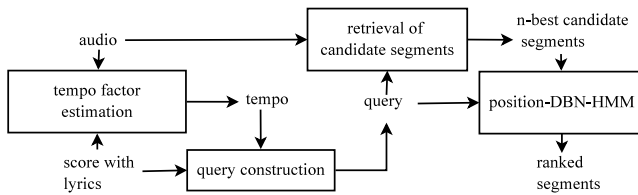


Figure 1. Approach overview: A key-phrase query is constructed in two variants: in the first stage candidate segments from audio are retrieved. In the second stage the query is modeled by a DBN-HMM aware of the position in music score. The DBN-HMM decodes and ranks candidate segments

utilized for detecting characteristic phrases (of 2-3 words) in sung audio. The method has been evaluated on polyphonic audio from Japanese pop, achieving 30% correctly identified links. Another modeling approach has been chosen in [1]. The authors propose subsequence dynamic time warping (SDTW) to find a match to an example utterance of a keyword as a subsequence of features from a target recording.

In summary, performance of the few works on lyrics spotting is not sufficiently good for practical applications. A probable reason for this is that hitherto approaches do not take into account the duration of syllables, which, as stated above, is an important factor that distinguishes speech from singing. In addition to that, syllable durations have been shown to be a strong reinforcing cue for the related task of automatically synchronizing lyrics and singing voice [3].

2.2 Position-aware DBN-HMMs

The modeling in most of the above mentioned approaches relies on HMMs. A drawback of HMMs is that their capability to model exact state durations is restricted, because the wait time in a state becomes implicitly an exponential distribution density [13,20, IV.D].

One alternative to tackle durations can be seen in dynamic Bayesian networks (DBN) [12], which allow modeling of interdependent musical aspects in terms of probabilistic dependencies. In [18] it was proposed how to apply DBNs to represent jointly tempo and the position in a musical bar as latent variables in a HMM. In a later work this idea was extended by explicitly modeling rhythmic patterns to track beats in music signals [7]. Relying on a similar DBN-based scheme, in [5] it has been shown, that the dependence of score position on structural sections makes it possible to link musical performances to score. In this paper for brevity we will refer to HMMs, which use DBNs to describe their hidden states, as DBN-HMMs.

3. APPROACH OVERVIEW

Figure 1 presents an overview of the proposed approach. A user first selects a query phrase from the lyrics of a composition of interest. Input are an audio recording, the queried lyrics and their corresponding excerpt from musical score.

Only recordings of performances of the composition of the query are being searched. Output is a ranked list of retrieved hit audio segments and their timestamps.

One of the common approaches to KWS in speech, known as acoustic KWS, is to decompose a keyword into acoustic phoneme models [16]. Similarly, in a first stage of our approach a SDTW retrieves a set of candidate audio segments that are acoustically similar to the phonemes-decomposed query.

In a second stage, durations of the query phonemes are modeled by a novel DBN-HMM (in short position-DBN-HMM). Tracking the position in music score, it augments the phoneme models with score reference durations. Next, we run a Viterbi decoding on each candidate segment separately. This assures that only one (the most optimal) path is detected for each candidate audio segment. Only full matches of the query are considered as hits and all hit results are ranked according to the weights derived from the Viterbi decoded path.

In what follows each of the two stages is described in details, preceded by remarks on tempo estimation and how a query key-phrase is handled.

3.1 Tempo factor estimation

Often a performance is not played at the tempo indicated in the score. To estimate a factor τ , by which the average tempo of the performance differs relative to the score tempo, we use the tonic-independent partial audio-score alignment methodology explained in [15]. The method uses Hough transform, a simple line detection method [2], to locate the initial section from score in the audio recording. We derive the tempo factor τ from the angle θ of the detected line (approximating the alignment path) in the similarity matrix between the score subsequence and the audio recording.

3.2 Query construction

A selected lyrical phrase serves as a query twice: first a *simple query* for retrieval of candidate segments and then a *duration-informed query* for the decoding with position-DBN-HMM.

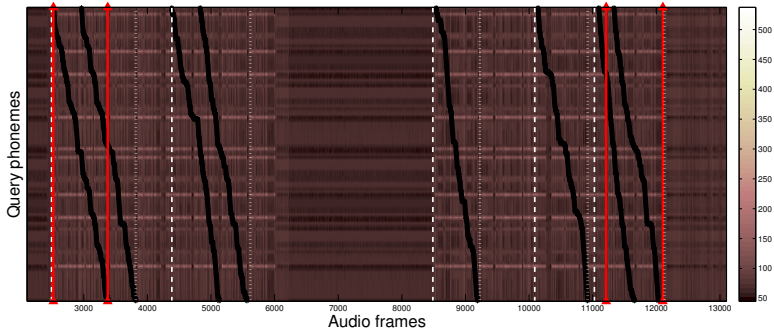


Figure 2. Distance matrix \mathcal{D} for an audio excerpt of around 100 seconds. Retrieved paths are depicted as black contours. White vertical lines indicate beginning (dashed) and ending (dotted) of candidate audio segments, whereas red lines with triangle markers surround the ground truth regions.

3.2.1 Acoustic features

For each of 38 Turkish phonemes (and for a silent pause model) a 3-state HMM is trained from a 5-hours corpus of Turkish speech [14]. The 3 states represent respectively the beginning, middle and ending acoustic state of a phoneme. The transition probabilities of the HMMs are not taken into account. The phoneme set utilized has been developed for Turkish and is described in [14]. The formant frequencies of spoken phonemes can be induced from the spectral envelope of speech. To this end, we utilize the first 12 MFCCs and their delta to the previous time instant, extracted as described in [19]. For each state a 9-mixture Gaussian distribution is fitted on the feature vector.

3.2.2 Simple query

For the first step no score-position information is utilized: lyrics is merely expanded to its constituent phoneme models. Let $\lambda_n \in \Lambda$ be a state of phoneme model at position n in the query, where Λ is a set of all 3×38 states for the 3 phonemes.

3.2.3 Duration-informed query

Unlike the simple query, a duration-informed query exploits the note-to-syllable mappings, present in sheet music. For each syllable a reference duration is derived by aggregating values of its associated musical notes. Then the reference durations are spread among its constituent phonemes in a rule-based manner, resulting in reference durations R_ϕ for each phoneme ϕ ¹.

To query a particular performance of a composition, R_ϕ are rescaled by the tempo factor τ (see section 3.1). Now this allows to define a mapping

$$f(p_n, s_n) \rightarrow \lambda_n \quad (1)$$

¹ In this work a simple rule is applied: consonants are assigned a fixed duration (0.1 seconds) and the rest of the syllable is assigned to the vowel.

that determines the true state λ_n from a phoneme network, being sung at position p_n within a section s_n . A position p_n can span the duration of a section $D(s_n) = \sum_{\phi \in s_n} R_\phi$.

4. RETRIEVAL OF CANDIDATE SEGMENTS

SDTW has proven to be an effective way to spot lyrics, in which the feature series of an audio query can be seen as a subsequence of features of a target audio [1]. In our case a query of phoneme models Λ with length M can be seen as subsequence of the series of MFCC features with length N , extracted from the whole recording. To this end we define a distance metric for an audio frame y_m and model state λ_n as a function of the posterior probability.

$$d(m, n) = -\log P(y_m | \lambda_n) \quad (2)$$

where for phoneme state model λ_n

$$P(y_m | \lambda_n) = \sum_{c=1}^9 w_{c, \lambda_n} \mathcal{N}(y_m; \mu_{c, \lambda_n}, \Sigma_{c, \lambda_n}) \quad (3)$$

with \mathcal{N} being the Gaussian distribution from a 9-component mixture with weights w_{c, λ_n} . Based on the distance metric 2 a distance matrix $\mathcal{D}^{N \times M}$ is constructed.

4.1 Path computation

Let a warping path Ω be a sequence of L points $(\omega_1, \dots, \omega_L)$, $l \in [1, L]$ and $\omega_l = (m, n)$ refers to an entry $d(m, n)$ in \mathcal{D} . Following the strategy and notation of [11] to generate Ω we select step sizes $\omega_l - \omega_{l-1} \in \{(1, 1), (1, 0), (1, 2)\}$ corresponding respectively to diagonal, horizontal and skip step. A horizontal step means staying in the same phoneme in next audio frame. The step size $(0, 1)$ is disallowed because each frame has to map to exactly one phoneme model. To counteract the preference for the diagonal and

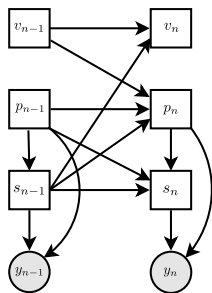


Figure 3. Representation of the hidden layers of the proposed model as a dynamic Bayesian network. Hidden variables (not shaded) are v - velocity, p - score position and s - section. The observed feature vector y is not shaded. Squares and circles denote respectively continuous and discrete variables

the skip step, we set rather high values for the local weights w_d and w_s [11].

A list of r candidate paths ($\Omega_1^*, \dots, \Omega_r^*$) is computed by iteratively detecting the current path with maximum score. After having detected a path Ω^* with final position in frame n^* a small region of 5% of M : ($n^* - 5\%M, n^* + 5\%M$) is blacklisted from further iterations, as described in [11]. This assures that the iterative procedure will not get stuck in a set of paths from a vicinity of a local maximum, but instead will retrieve as many relevant audio segments as possible.

4.2 Candidate segment selection

Analysis of the detected query segments revealed that a path often matches only partially the correct section segment. However, usually different parts of a segment have been detected in neighbouring paths. To handle this, we consider candidate segments - segments from the target audio, within which a frame y_m belongs to more than one path Ω . In other words, a candidate segment spans audio from the initial timestamp of the leftmost path to the final timestamp of the rightmost path. An example of retrieved candidate segments is presented in Figure 2. It can be seen that the two ground truth regions lie within candidate segments, which consist of more than one path.

5. POSITION-DBN-HMM

In this section we present the novel position-DBN-HMM for modeling a lyrical phrase. Its main idea is to incorporate the phonetic identities of lyrics and the syllable durations, available from musical score, into a coherent unit. The dependence of the observed MFCC features (that capture the phonetic identity) on musical velocity and score position are presented as DBN in Figure 3.

5.1 Hidden variables

1. Position p_n from musical score for a section ($p_n \in \{1, \dots, D(s_n)\}$). $D(s_n = Q)$ is the total duration for a section s_n as defined in section 3.2.3. Note that $D(s_n)$ for a given section is different for two performances with different tempo, because of the tempo factor τ .
2. Velocity $v_n \in \{1, 2, \dots, V\}$. Unit is the number of score positions per audio frame. Staying in state $v_n = 2$, for example, means that the current tempo is steady and around 2 times faster than the slowest one.
3. Structural section $s_n \in \{Q, F\}$ where Q is the queried section and F is a filler section. A filler section represents any non-key-phrase audio regions, and practically allows with equal probability being in any phoneme state (see section 5.3)

We compensate for tempo deviations by varying the local step size of the v variable. To allow handling deviations of up to half tempo, the derived $D(s_n = Q)$ is multiplied by 2. This means that $v = 1$ corresponds to half of the detected tempo. For the experiments reported in this paper, we chose $V = 5$. Furthermore we set $D(s_n = F) = V$. This assures that even in fastest tempo there is an option of entering the filler section.

The proposed model is different from the model proposed in [7] in two aspects:

- $D(s_n = Q)$ is not fixed but depends on the section of interest and the detected tempo of performance
- a section s_n (a pattern in the original model) is not fixed, but can vary between a query and filler states $\{Q, F\}$

Since all the hidden variables are discrete, one can reduce this model to a regular HMM by merging all variables into a single 'meta-variable' x_n :

$$x_n = [v_n, p_n, s_n] \quad (4)$$

Note that the state space becomes the Cartesian product of the individual variables.

5.2 Transition model

Due to the conditional independence relations presented in Figure 3, the transition model reduces to

$$\begin{aligned} P(x_n | x_{n-1}) = & P(v_n | v_{n-1}, s_{n-1}) \times \\ P(p_n | v_{n-1}, p_{n-1}, s_{n-1}) \times & P(s_n | p_{n-1}, s_{n-1}, p_n) \end{aligned} \quad (5)$$

5.2.1 Velocity transition

$$p(v_n | v_{n-1}) = \begin{cases} \phi/2, & v_n = v_{n-1} \pm 1 \\ 1 - \phi, & v_n = v_{n-1} \\ 0, & \text{else} \end{cases} \quad (6)$$

where ϕ is a constant probability of change in velocity and is set to 0.2 in this work.

5.2.2 Position transition

The score position is defined deterministically according to:

$$p_n = (p_{n-1} + v_{n-1} - 1) \quad \text{mod } D(s_{n-1}) + 1 \quad (7)$$

where the modulus operator resets the position to be in a beginning of a new section after it exceeds the duration of previous section $D(s_{n-1})$

5.2.3 Section transition

$$P(s_n | p_{n-1}, s_{n-1}, p_n) = \begin{cases} P(s_n | s_{n-1}), & p_n \leq p_{n-1} \\ 1, & p_n > p_{n-1} \ \& \ s_n = s_{n-1} \end{cases} \quad (8)$$

A lack of increase in the position is an indicator that a new section should be started. $P(s_n | s_{n-1})$ is set according to a transition matrix $A = \{a_{ij}\}$ where $i \in \{Q, F\}$ and self transitions a_{QQ} and a_{FF} for query and filler section respectively can be set to reflect the expected structure of the target audio signal. In this work we set $a_{QQ} = 0$, since we expect that a query might be decoded at most once in a candidate audio segment. The value $a_{FF} = 0.9$ is determined empirically.

5.3 Observation model

For the query section the probability of the observed feature vector in position p_n from section s_n is computed for the model state λ_n by a mapping function $f(p_n, s_n)$, introduced in section 3.2. A similar mapping function has been proposed for the first time in the DBN-HMM in [5].

Then

$$P(y_n | p_n, s_n = Q) = P(y_n | \lambda_n) \quad (9)$$

which reduces to applying the distribution defined in Equation 3.

In case of the filler section the most likely phoneme state is picked.

$$P(y_n | p_n, s_n = F) = \max_{\lambda \in \Lambda} P(y_n | \lambda) \quad (10)$$

Note that position p_n plays a role only in tracking the total section duration $D(s_n = F)$.

5.4 Inference

An exact inference of the 'meta-variable' x can be performed by means of the Viterbi algorithm. A key-phrase is detected whenever a segment of the Viterbi path Ω passes through a section $s_n = Q$. The likelihood of this path segment is used as detection score for ranking all retrieved key-phrases.

6. DATASET

The test dataset consists of 12 a-cappella performances of 11 compositions with total duration of 19 minutes.

statistic	value
#section queries	50
average cardinality \bar{C}_q	3.2
maximum cardinality C_{qM}	6
#words per section	5-14
#sections per recording	6-16
#phonemes per section	26-63

Table 1. Statistics about queries (lyrics sections with unique lyrics) in the test dataset. The low value of \bar{C}_q are due to the small number of performances per composition.

The compositions are drawn from the CompMusic corpus of classical Turkish Makam repertoire [17]. The a-cappella versions have been sung by professional singers and recorded especially for this study. Scores are provided in the machine-readable symbTr format [6], which contain marks of section divisions. A performance has been recorded in-sync with the original recording, whereby instrumental sections are left as silence. This assures that the order, in which sections are performed, is kept the same².

We consider as a query q each section from the scores, which has unique lyrics: in total 50. Note that the search space is restricted to all recordings of the composition, from which the section is taken. In a given recording we annotated the section boundary timestamps. Let C_q be the total number of relevant occurrences (cardinality) of a query q . Table 1 presents the average cardinality \bar{C}_q and other relevant statistics about sections.

7. EVALUATION

7.1 Evaluation metrics

Having a ranked list of occurrences of each lyrical query, the search-by-lyrics can be interpreted as a ranked retrieval problem, in which the users are interested in checking only the top K relevant results [10]. This allows to reject irrelevant results by considering only top K results in the evaluation metric. We consider this strategy as appropriate since a query has low average cardinality ($\bar{C}_q = 3.2$). Let the relevance of ranked results for a query q be $[r_q(1), \dots, r_q(n_q)]$ where n_q is the number of retrieved occurrences. Note that a detected audio segment is either hit or not, making $r_q(k) \in \{0, 1\}$.

For each of the queried score sections an average precision P_q at different values of K is computed as:

$$\bar{P}_q = \frac{1}{C_q} \sum_{k=1}^K r_q(k) P_q(k) \quad (11)$$

as defined in [10], where $P_q(k)$ is precision at k . The relevance $r_q(k)$ of k^{th} retrieved occurrence is binary and set to 1 only if both retrieved boundary timestamps are within a tolerance window of 3 seconds from ground truth. This window size has been introduced in [9] and is commonly used for evaluating structural segments. The hits are

²The dataset is available here: <http://compmusic.upf.edu/turkish-makam-acappella-sections-dataset>

K	1	2	3	4	5	6
<i>SDTW</i>	8.3	12.1	16.2	19.0	22.0	25.7
<i>DBN-HMM</i>	5.0	7.7	18.75	28.8	35.0	37.9

Table 2. MAPs (in percent) for ranked result segments for two system variants: baseline with SDTW and complete with position-DBN-HMM.

ranked by the likelihoods of the relevant Viterbi path segments. Results are reported in terms of mean average precision (MAP) as the average over all \bar{P}_q .

7.2 Experiments

To assess the benefit of the proposed modeling of positions, we conduct a comparison of the performance of the complete system and a baseline version without the position-DBN-HMM³. For the baseline, as result set we consider the audio segments corresponding to the list of candidate paths ($\Omega_1^*, \dots, \Omega_r^*$) derived after SDTW (see section 4.1). As a ranking strategy, SDTW-paths are ordered by means of the sum of distance metrics $d(m, n)$, which is derived from the observation probability. We report results at different values for K in Table 2. Results for $K > C_{qM}$ are omitted. Furthermore, we picked empirically $r = 12$ candidate paths in SDTW, which is twice C_{qM} .

The results confirm the expectation that the performance of SDTW alone is inferior. Retrieving relevant candidate paths seemed to be very dependent on the weights w_d and w_s for the diagonal and skip steps. We noted that adapting weights for a recording according to the detected tempo factor τ might be beneficial, but did not conduct related experiments in this work. The optimal values ($w_d = 6.5$ and $w_s = 11$) in fact guaranteed good coverage of relevant segments in the slowest tempo in the dataset.

As K increases, the MAP for both DBN-HMM and SDTW improves, as more hits are being found on lower ranks. However top ranks are relatively low for DBN-HMM. This indicates that the Viterbi weighting scheme might not be optimal. In general, MAP for DBN-HMM, at higher values of K gets substantially better than the baseline, which suggests that modeling syllable durations is beneficial. A further reason might be that the position-DBN-HMM can model tempo in a more flexible way and is thus not affected by the difference between the tempo indicated in the score and the real performance tempo.

7.3 Comparison to related work

For the sake of comparison to any future work we report in Table 3 the f-measure, derived from the precision $P_q(k)$ and recall $R_q(k)$ as defined in [10]. Unfortunately, no direct comparison to previous work on lyrics spotting [1, 4, 8] is possible, because these works rely on speech models for languages different from Turkish. Furthermore, the evaluation setting in none of the works is comparable to ours.

³ To facilitate reproducibility of this research source code is publicly available here: <https://github.com/georgid/Position-DBN-HMM-Lyrics>

K	1	2	3	4	5	6
<i>DBN-HMM</i>	12.4	15.5	19.2	24.2	31.3	37.8

Table 3. F-measure (in percent) for the position-DBN-HMM for ranked results segments

In [8] a result is considered true positive if a keyword is detected at any position in an expected audio clip. The authors argue that since a clip spans one line of lyrics (only 1 to 10 words) this is sufficiently exact, whereas we are interested in detecting the exact timestamps of a key-phrase. In addition to that, their longest query has 8 phonemes, which is much less than the average in our setting.

In [4] the accuracy of the key-phrase spotting module is not reported, but instead only the percentage of the correctly detected links connecting key-phrases from a song to another song. It can be inferred from it that an upper bound on the performance of the key-phrase spotting lies around an accuracy of 30%. Further, on creating a link for a given key-phrase only the candidate section with highest score for a song has been considered, which might ignore any other true positives.

8. CONCLUSION

In this study we have investigated an important problem that has started to attract attention of researchers only recently. We tackle the linking between audio and structural sections from the perspective of lyrics: we proposed a method for searching in musical audio for the occurrences of a characteristic section-long lyrical phrase. We presented a novel DBN-based HMM for tracking sung phoneme durations. Evaluation on a-cappella material from Turkish Makam music shows that the search with the proposed model brings substantial improvement compared to a baseline system, unaware of syllable durations.

We plan to focus future work on applying the proposed model to the case of polyphonic singing. We expect further, that this work can serve as a baseline for further research on singing material with similar musical characteristics.

We want to point as well that, the proposed score-informed scheme is applicable not necessarily only when musical scores are available. Scores can be replaced by any format, from which duration information can be inferred: for example annotated melodic contour or singer-created indications along the lyrics.

Acknowledgements This work is partly supported by the European Research Council under the European Union's Seventh Framework Program, as part of the CompMusic project (ERC grant agreement 267583) and partly by the AGAUR research grant.

9. REFERENCES

- [1] Christian Dittmar, Pedro Mercado, Holger Grossmann, and Estefania Cano. Towards lyrics spotting in the

- syncglobal project. In *Cognitive Information Processing (CIP), 2012 3rd International Workshop on*, pages 1–6. IEEE, 2012.
- [2] Richard O Duda and Peter E Hart. Use of the Hough transformation to detect lines and curves in pictures. *Communications of the ACM*, 15(1):11–15, 1972.
- [3] Georgi Dzhabazov and Xavier Serra. Modeling of phoneme durations for alignment between polyphonic audio and lyrics. In *Sound and Music Computing Conference*, Maynooth, Ireland, 2015.
- [4] Hiromasa Fujihara, Masataka Goto, and Jun Ogata. Hyperlinking lyrics: A method for creating hyperlinks between phrases in song lyrics. In *Proceedings of the 9th International Conference on Music Information Retrieval*, pages 281–286. Philadelphia, USA, September 14-18 2008.
- [5] Andre Holzapfel, Umut Şimşekli, Sertan Şentürk, and Ali Taylan Cemgil. Section-level modeling of musical audio for linking performances to scores in turkish makam music. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brisbane, Australia, 19/04/2015 2015.
- [6] M Kemal Karaosmanoğlu. A Turkish makam music symbolic database for music information retrieval: Symbtr. In *Proceedings of the 13th International Society for Music Information Retrieval Conference*, Porto, Portugal, 2012.
- [7] Florian Krebs, Sebastian Böck, and Gerhard Widmer. Rhythmic pattern modeling for beat and downbeat tracking in musical audio. In *Proceedings of the 14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil, November 4-8 2013.
- [8] Anna M. Kruspe. Keyword spotting in a-capella singing. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, pages 271–276, Taipei, Taiwan, 2014.
- [9] Mark Levy and Mark Sandler. Structural segmentation of musical audio by constrained clustering. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(2):318–326, 2008.
- [10] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA, 2008.
- [11] Meinard Müller. *Information retrieval for music and motion*, volume 2. Springer, 2007.
- [12] Kevin Patrick Murphy. *Dynamic bayesian networks: representation, inference and learning*. PhD thesis, University of California, 2002.
- [13] Lawrence Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.
- [14] Özgül Salor, Bryan L. Pellom, Tolga Ciloglu, and Mbeccel Demirekler. Turkish speech corpora and recognition tools developed by porting sonic: Towards multilingual speech recognition. *Computer Speech and Language*, 21(4):580 – 593, 2007.
- [15] Sertan Şentürk, Sankalp Gulati, and Xavier Serra. Score informed tonic identification for makam music of turkey. In *Proceedings of 14th International Society for Music Information Retrieval Conference*, pages 175–180, Curitiba, Brazil, 2013.
- [16] Igor Szöke, Petr Schwarz, Pavel Matejka, Lukáš Burget, Martin Karafiát, Michal Fapso, and Jan Cernocký. Comparison of keyword spotting approaches for informal continuous speech. In *Interspeech*, pages 633–636, 2005.
- [17] Burak Uyar, Hasan Sercan Atlı, Sertan Şentürk, Barış Bozkurt, and Xavier Serra. A corpus for computational research of Turkish makam music. In *1st International Digital Libraries for Musicology Workshop*, pages 57–63, London, United Kingdom, 2014.
- [18] Nick Whiteley, A. Taylan Cemgil, and Simon Godsill. Bayesian modelling of temporal structure in musical audio. In *Proceedings of the 7th International Conference on Music Information Retrieval*, Victoria (BC), Canada, October 8-12 2006.
- [19] Steve J Young. *The HTK hidden Markov model toolkit: Design and philosophy*. Citeseer, 1993.
- [20] Shun-Zheng Yu. Hidden semi-markov models. *Artificial Intelligence*, 174(2):215–243, 2010.

QUANTIFYING LEXICAL NOVELTY IN SONG LYRICS

Robert J Ellis, Zhe Xing, Jiakun Fang, and Ye Wang

School of Computing, National University of Singapore

{robjellis, xingzhe.cs}@gmail.com; {fangjiak, wangye}@comp.nus.edu.sg

ABSTRACT

Novelty is an important psychological construct that affects both perceptual and behavioral processes. Here, we propose a lexical novelty score (LNS) for a song's lyric, based on the statistical properties of a corpus of 275,905 lyrics (available at www.smcnus.org/lyrics/). A lyric-level LNS was derived as a function of the inverse document frequencies of its unique words. An artist-level LNS was then computed using the LNSs of lyrics uniquely associated with each artist. Statistical tests were performed to determine whether lyrics and artists on Billboard Magazine's lists of "All-Time Top 100" songs and artists had significantly lower LNSs than "non-top" songs and artists. An affirmative and highly consistent answer was found in both cases. These results highlight the potential utility of the LNS as a feature for MIR.

1. INTRODUCTION

From 2004 through 2013, both U.S. and worldwide Google searches for "lyrics" outnumbered searches for "games", "news", and "weather", as computed by Google Trends¹. The importance listeners place on song lyrics has motivated several explorations for translating a song's lyric into queryable features: for example, by topic [1], genre [2], or mood [3–6]. All these cited examples have incorporated *word frequency* information: as a key statistic in the computational process. The inverse document frequency (IDF) statistic, for example, is used to identify "diagnostic" terms within a lyric that can be further related to a particular topic, genre, or mood.

In the present paper, we propose using IDF information to derive a quantifiable and queryable feature of song lyrics: a *lexical novelty score* (LNS). "Lexical" refers to properties of individual words, as distinct from their grammatical function or syntactical arrangement. Our LNS is based, in part, on the trimean of IDFs associated with the set of unique words in a lyric. The greater the number of statistically infrequent (i.e.,

"novel") words in a lyric, the higher its IDF trimean.

Why might such a quantification of lexical novelty be useful? A number of answers emerge from the domains of psycholinguistics and psychology. The novelty or unfamiliarity of a stimulus has a direct bearing on basic cognitive processing. For example, words that are statistically infrequent (i.e., have a high IDF) are more difficult to perceive, recognize, and recall than more commonly encountered words (e.g., [7–9]). The affective response *associated* with perceiving novelty, however, is a more complex process. Berlyne [10], for example, extended a classic *inverted-U* relationship first proposed by Wilhelm Wundt [11]: a peak level of perceived pleasantness or "hedonic value" for moderately complex or moderately novel stimuli, and decreased liking for very simple/familiar or very complex/novel stimuli. Such a relationship has been documented across numerous classes of stimuli, including music [12], and can be further modified by an individual perceiver's *preferences* for novelty—a construct that has informed influential models of human personality [13].

Taken together, this evidence suggests that a method to quantify novelty/complexity within song lyrics might find application within the domain of personalized music recommendation. First, generated playlists could be optimized with the "right" level of lyric complexity based upon the user's activity state (e.g., exercising, commuting, or intense studying) [14–15]. Second, by computing the level of lexical novelty in a user's favorite artist, novel artists with a similar level of lexical novelty could be recommended. Third, songs with lyrics that are "not-too-simple" or "not-too-complex" could be used in paradigms supporting native or second language learning [16–17] or language recovery after brain injury [18].

2. RELATED WORK

Methods for translating a text into a single summary statistic or "grade" have been employed in a number of domains. Mid-twentieth century development of *readability metrics*—designed to quantify the ease with which a written text could be comprehended—emerged from the human factors literature (for a review and some context, see [19]), and have come to be widely applied in a variety of natural language settings [20–21]. Readability metrics are simple mathematical transformations of a text's orthographic features: letter



© Robert J Ellis, Zhe Xing, Jiakun Fang, and Ye Wang.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Robert J Ellis, Zhe Xing, Jiakun Fang, and Ye Wang. "Quantifying lexical novelty in song lyrics", 16th International Society for Music Information Retrieval Conference, 2015.

¹ <http://www.google.com/trends/?explore#q=lyrics,+games,+news,+weather&cmpt>

count, syllable count, word count, and sentence count.² Word frequency information is only rarely incorporated into readability calculations; for example, tallying the number of “difficult” [22] or “unfamiliar” [23] words (as defined by a set of 3000 words), or the “average grade level” of words (from a set of 100,000 words) [24].

By contrast, word frequency information is fundamental to vector space model approaches for text retrieval [25]. The process by which candidate documents are matched to a particular query often involves the use of *term frequency-inverse document frequency* (tf*idf) calculations [26–27]. A useful summary statistic across a set of query terms is their average IDF [28–29]. It should be noted that our proposed idea of a lexical novelty score is distinct from prior uses of tf*idf for *novelty detection* [30], which attempts to detect new information in a “stream” of documents. It is also distinct from *acoustic novelty* audio segmentation methods based on changes in temporal self-similarity [31].

3. DATASETS AND PREPROCESSING STEPS

3.1 Word frequency tables

The two key data sources for the proposed IDF-based lyric LNS are a lyrics corpus and a look-up table of document frequencies (DFs). Word frequencies *could* be estimated from the lyrics corpus itself. However, such an operation could create a dependency between IDFs and resultant LNSs—or at least necessitate retabulating word frequencies and IDFs as more lyrics were added to the corpus. Word frequency values derived from an *independent* corpus were thus desirable.

Numerous tables of word frequencies have been published (reviewed in [32]): for example, the Brown corpus (1 million words), British national corpus (100M words), Corpus of Contemporary American English (450M words), and Google Books corpus (155 billion words of American English). In the present work, we selected the use word frequency tables derived from the SUBTLEX_{US} corpus [9]; a corpus of subtitle transcripts of 8388 American films and television programs. A list of 74,286 non-stemmed words⁵ (46.7M word instances in total) has been compiled, with DFs (from 1 to 8388) and corpus frequencies (from 1 to 2,134,713) tabulated for each word. In addition to being fully and freely available⁶, SUBTLEX_{US} word frequencies have the appealing property of being derived from *spoken* source material, which may provide a closer match to the usage patterns in *sung* speech. The IDF of the *i*th word in the SUBTLEX_{US} table was computed as $\log_{10}(8388/DF_i)$.

3.2 Lyrics corpus

Next, we discuss the issue of an appropriate lyrics corpus. The Million Song Dataset [33] is associated with a smaller lyrics corpus (237,662 lyrics)⁷, obtained in partnership with musiXmatch⁸. The bag-of-words format used to store each lyric, however, only references the 5000 most frequent word *stems* (the part of a word common to all its inflectional and derivational variants; for example, “government”, “governor”, “governing”, and “governance” are all stemmed to “govern”) as computed by the Porter2 stemmer⁹. (In fact, the 5000-item stemmed word list contains more than 1000 non-English stems when cross-checked with a 266,447-item dictionary derived from existing dictionary lists¹⁰.) The manner in which word variants are used during communication, however, conveys rich information about the communicator’s language facility [35–37]. Furthermore, word variants can have very different IDFs; in SUBTLEX_{US}, the four variants of “govern” listed above have IDFs of .74, 1.32, 2.58, and 3.22, respectively. As a result, a LNS derived from word stems would ignore potentially “diagnostic” differences in lexical usage between lyrics.

For this reason, a new lyrics corpus was obtained via special arrangement with LyricFind¹¹, a leading provider of legal lyrics licensing and retrieval. In addition to the lyrics corpus itself, metadata comprising performing artist, album, lyricist, and license territory information for each lyric was made available. The full corpus contained 587,103 lyrics. After restricting the corpus to lyrics with United States copyright, 389,029 lyrics remained.

3.3 Lyrics pre-processing

A multi-step procedure converted each lyric from its original text format into a bag-of-words format. Each lyric was first “cleaned” using a series of hand-crafted transformation rules (i.e., $x \rightarrow x'$): (1) splitting of compounds (e.g., *half-hearted*→*half hearted*) or removal of hyphenated prefixes (e.g., *mis-heard*→*misheard*); (2) elimination of contractions (e.g., *you’ll ve*→*you will have*; *gonna*→*going to*); (3) restoration of dropped initial (e.g., *’til*→*until*), interior (e.g., *ne’er*→*never*), or final (*tryin’*→*trying*) letters; (4) abbreviation elimination (e.g., *mr.*→*mister*); (5) adjustment of British English to American English spellings (e.g., *colour*→*color*)¹²; and (6) correction of 4264 commonly misspelled words¹³.

Each lyric was then cross-checked with the 266,447-item dictionary. Lyrics in which fewer than 80% of

⁷ <http://labrosa.ee.columbia.edu/millionsong/musixmatch>

⁸ www.musixmatch.com

⁹ <http://snowball.tartarus.org>

¹⁰ <http://wordlist.aspell.net>

¹¹ www.lyricfind.com

¹² Using <http://wordlist.aspell.net/varcon>

¹³ Using http://en.wikipedia.org/wiki/Wikipedia:Lists_of_common_misspellings/For_machines

² For an illustration, www.readability-score.com

⁵ The following items in the SUBTLEX_{US} table were excluded from this tally: ‘d, ‘s, ‘m, ‘t, ‘ll, ‘re, dot, gonna, wanna, couldn, did, doesn.

⁶ <http://expsy.ugent.be/subtlexus/>

unique words could be matched to the dictionary were eliminated; 360,919 lyrics remained. After removing duplicate lyrics, the final corpus contained 275,905 lyrics.

A total of 67.6M word instances was present in this set of songs, with 66,975 unique words. Of these items, 51,832 were an exact match with the 74,286-item SUBTLEX_{US} word list; this accounted for 99.7% of the 67.6M word instances in the lyrics corpus. IDFs derived from the SUBTLEX_{US} corpus were generally in agreement with IDFs derived from the LyricFind corpus itself (Pearson's $r = .837$).

4. LYRIC-LEVEL LEXICAL NOVELTY SCORE

4.1 First-pass LNS: IDF_{TM}

A first-pass LNS for a lyric was defined as the trimean of SUBTLEX_{US}-derived IDFs (IDF_{TM}) associated with the set of w unique words in that lyric (w_u):

$$TM = \frac{Q_1 + 2Q_2 + Q_3}{4}, \quad (1)$$

where Q_1 , Q_2 , and Q_3 are the first quartile, second quartile (median), and third quartile, respectively. The trimean is an outlier-robust measure of central tendency [37]. For example, a low-frequency variant of a common word not “corrected” during the cleaning step would yield a spuriously high IDF; the trimean (but not the arithmetic mean) is robust to this kind of outlier.

The higher a lyric's IDF_{TM} , the more low-frequency (i.e., novel) words it contains. Figure 1 plots IDF_{TM} as a function of w_u for all 275,905 lyrics (using \log_{10} scaling on the x-axis). Observed w_u values range from 12 to 895.

A few illustrative cases are highlighted on Figure 1. The highest IDF_{TM} ($= 2.3212$; LyricID 1142131; marked ①) is “Yakko’s World” from the cartoon *Animaniacs*. (Example text: “There’s Syria, Lebanon, Israel, Jordan / Both Yemens, Kuwait, and Bahrain / The Netherlands, Luxembourg, Belgium, and Portugal / France, England, Denmark, and Spain”.) The lowest IDF_{TM} ($= 0.0016$; LyricID 53540; marked ②) is “You Don’t Know” by Killing Heidi. (“I can see you / And you don’t have a clue / Of what you’ve done / And there’s no reason / For what you’ve done to / Done to my ...”)

Lyric ③ (LyricID 786811; “One More Bite of the Apple” by Neil Diamond) has the same w_u as ① ($= 153$), but a much lower IDF_{TM} ($= 0.0804$), indicating lower lexical novelty: “Been away from you for much too long / Been away but now I’m back where I belong / Leave while I was gone away / But I do just fine”. Lyric ④ (LyricID 78427; “Revelation” by Blood) has nearly the same w_u as ② (24 vs. 23) but a much higher IDF_{TM} ($= 1.5454$), indicating higher lexical novelty (“Writhe and shiver in agonies undreamable / Wriggling and gasping / Anticipating the tumescent / Revelation of the flesh”).

Finally, cases ⑤ (LyricID 335431; “The Tear Drop”

by Armand van Helden) and ⑥ (LyricID 1452671; “Sunshine” by Bow Wow) both have $w_u = 195$, but very different IDF_{TM} values (1.8464 vs. 0.1378). High lexical novelty is present in ⑤ (“A buttress breaching barrage blast / A tumultuous thunderbolt tirade / An annihilating eradicating avalanche of absolute absoletion”); low lexical novelty is present in ⑥ (“What you hear me talkin’ ’bout / You just ain’t gonna find out / Walkin’ around in somebody’s club / Now she’s sayin’ her house”).

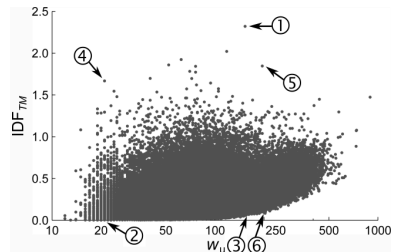


Figure 1. Scatter plot of unique words (w_u) versus IDF_{TM} .

A clear relationship is visible between w_u and IDF_{TM} (Pearson's $r = .477$): as w_u increases, so does the minimum observed IDF_{TM} . This can be attributed to statistical patterns present in natural language. Specifically, a small number of words account for a large percentage of total word instances; a phenomenon which follows Zipf's law (e.g., [38]). In the SUBTLEX_{US} corpus, for example, 10 words (*you, i, the, to, a, it, that, and, of, what*) account for 24.3% of all 46.7M word instances. Because IDF_{TM} is derived from the set of *unique* words in a lyric, as w_u increases, so too must the number of lower-frequency (i.e., higher-IDF) words, causing the IDF_{TM} to rise. Such a pattern would manifest for any L-estimator (mean, median, midhinge, etc.).

A more informative statistic could be obtained if the IDF_{TM} of a lyric with w unique words were compared against a large distribution of simulated IDF_{TM} values obtained from repeated random draws of w unique words from the set of lyrics that had more than w unique words. This procedure is formalized next.

4.2 Scaling IDF_{TM} : Monte Carlo simulations

Consider two lyrics, one with $IDF_{TM} = 0.25$ and $w_u = 50$, and the other with $IDF_{TM} = 0.5$ and $w_u = 200$. Two scaling distributions of simulated IDF_{TM} values were created using a 10,000-iteration procedure. To create the scaling distribution for $w_u = 50$, on each iteration, a single lyric was randomly selected from the set of 239,225 lyrics with $w_u > 50$. The full set of words in that lyric (including repeated words) was randomly permuted, the first 50 unique words pulled, and the IDF_{TM} of those words was taken. To create the scaling distribution for $w_u = 200$, a similar procedure was performed, using the set of 15,124 lyrics with $w_u > 200$. Figure 2 presents an

empirical cumulative distribution function (ECDF) of these two scaling distributions. The “scaled IDF_{TM} ” is defined as the percentile P (i.e., the y -axis value on the ECDF, multiplied by 100) where $x = IDF_{TM}$. In the above example, when $IDF_{TM} = 0.25$ and $w_u = 50$, $P = 85.8$. By contrast, when $IDF_{TM} = 0.25$ and $w_u = 200$, $P = 10.3$. This can be interpreted as follows: with a longer lyric ($w_u = 200$ vs. $w_u = 50$), the likelihood of obtaining an $IDF_{TM} > 0.5$ by chance (i.e., $100 - P$) is much higher (89.7% vs. 14.2%); that is, it is a *less novel* occurrence.

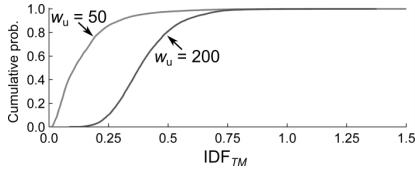


Figure 2. ECDFs of simulated IDF_{TM} values for two representative values of w_u .

To scale the full set of IDF_{TM} values, the above simulation was modified in the following manner. First, the range of target w_u values was capped at 275, thus reserving 5228 lyrics with $w_u > 275$ to create the scaling distribution for $w_u = 275$. Second, the set of target P -values was defined as .01 to 99.99 in increments of .01. Third, to accurately estimate the “tails” of P (i.e., values near 0 and 100), many more Monte Carlo iterations at each w_u are needed; thus, the number of iterations was increased from 10,000 to 1 million.

Figure 3 highlights the results of this simulation. A representative set of “iso-probability curves” resulting from the Monte Carlo simulation are superimposed on the scatter plot first shown in Figure 1. A given curve plots the P th percentile (where $P = \{.01, 10, 50, 90, 99, 99.9, 99.99\}$) of simulated IDF_{TM} values across the set of w_u values. $IDF_p \approx 0$ indicates *very low* lexical novelty, $IDF_p \approx 50$ indicates *moderate* lexical novelty, and $IDF_p \approx 100$ indicates *very high* lexical novelty. As expected, the iso-probability curves for low P -values mirror the pattern in the real data: higher IDF_{TM} values as w_u increases.

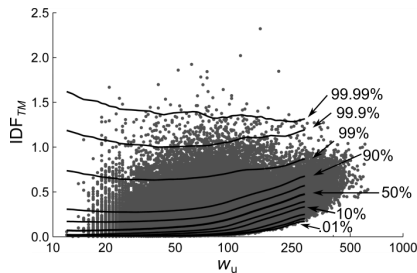


Figure 3. Representative iso-probability curves.

4.3 Second-pass LNS: Percentiles

Each IDF_{TM} was mapped to its corresponding IDF_p using nearest neighbor interpolation. IDF_{TM} values below $P = .01$ ($n = 80$) or above $P = 99.99$ ($n = 52$) were set to $IDF_p = 0$ or $IDF_p = 100$, respectively. Figure 4 plots IDF_p as a function of w_u for the final set of 270,677 unique lyrics. The relationship between w_u and IDF_p ($r = -.106$) is much weaker than between w_u and IDF_{TM} ($r = .477$). IDF_p values were roughly uniform (mean = 44.29; standard deviation = 29.70; skewness = 0.255).

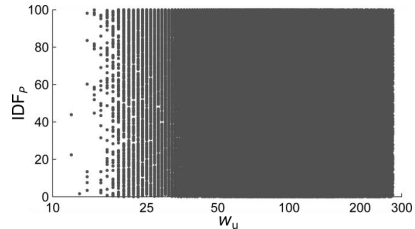


Figure 4. Percentile-transformed novelty scores (IDF_p) as a function of w_u .

Figure 5 presents an ECDF of both IDF_{TM} and IDF_p , highlighting the six lyrics discussed earlier. Compared to IDF_{TM} , IDF_p better differentiates lyrics with high lexical novelty (cases ①, ②, and ③) versus low novelty (cases ④, ⑤, and ⑥).

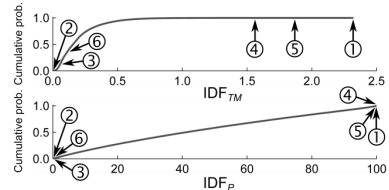


Figure 5. ECDFs for IDF_{TM} (upper) and IDF_p (lower).

5. ARTIST-LEVEL LEXICAL NOVELTY

Having defined IDF_p as the lyric-level LNS, we next sought to characterize lexical novelty at the artist level. Artist information was obtained via LyricFind ArtistIDs, which are distinct for different combinations of individual artists. To increase the specificity of an artist-level score, lyrics recorded by multiple artists (e.g., holiday songs, jazz standards) were excluded. Artists associated with fewer than 10 unique lyrics (λ_u) were deemed to have an insufficient catalog, and were ignored. A final set of 5884 artists (a total of 216,072 lyrics) remained. The trimean of each artist’s λ_u IDF_p values was then taken as a simple and intuitive artist-level LNS.

Figure 6 plots artist-level LNS as a function of λ_u ; no correlation was present between them ($r = -.009$). The distribution of values (mean = 43.49; standard deviation = 21.20) was roughly symmetrical (skewness = .459).

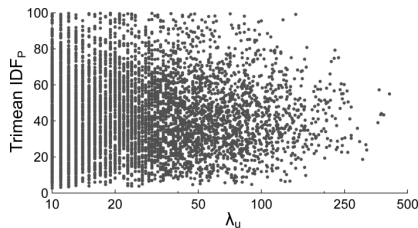


Figure 6. Artist-level LNS as a function of λ_u .

6. BILLBOARD MAGAZINE “TOP” LISTS

Having derived both a lyric-level and an artist-level point estimate of lexical novelty, any number of subsequent analyses may be performed. As an illustrative example, we turn to Billboard Magazine’s 2013 ranking of the “All-Time Top 100 Songs”¹⁴ and “All-Time Top 100 Artists”¹⁵. Rankings were calculated based on overall success on the magazine’s “Hot 100” chart, a weekly ranking of the top 100 popular music singles in the United States, published since August 1958 [40–41].

The Top Songs list was determined by Billboard using an inverse point system, with time spent in the #1 position of each weekly chart weighted highest, and time spent in the #100 position weighted lowest. Of the 100 songs on the list, 95 were present in the LyricFind corpus. Lyrics for the remaining five were queried from metrolyrics.com and processed as described in Section 4.

The Top Artists list was determined by Billboard by aggregating all the songs which charted over the course of each artist’s career. Of the 100 artists, 98 were among the set of 5884 artists with a valid artist-level LNS; the other two artists had $\lambda_u < 10$.

7. EXPERIMENTAL HYPOTHESES

Two hypotheses were examined, both driven by the assumption that high lexical novelty is less likely to be “chart-worthy”. Specifically, we predicted that both lyric-level and artist-level LNSs would be *lower* in the set of Top Songs and Top Artists relative to “non-top” songs and artists in the LyricFind corpus.

Statistical significance was assessed using a nonparametric two-sample Mann–Whitney (MW) test. A special sampling procedure was implemented to counteract the bias towards smaller p -values when comparing large samples [41]. On each of 10,000 iterations, two samples were drawn. The first sample was always the n Top Song or Top Artist LNSs, and the second sample was a random draw (without replacement) of n LNSs

from the remaining set of songs or artists (where n is 100 for songs and 98 for artists). The distribution of Z -values from the 10,000 MW tests indicates the strength of the difference between the samples: the more negative it falls, the greater our confidence that lexical novelty is *systematically* lower in the set of Billboard items.

8. EXPERIMENTAL RESULTS

8.1 Billboard Top Songs analysis

Figure 7a shows the ECDFs of lyric-level LNS for the set of 100 Top Songs and the remaining 270,582 songs. They are markedly different: LNSs for the Top Songs are “pulled” towards zero, indicating reduced lexical novelty in this set. Consistent with this, the distribution of Z -values (Figure 7b) is strongly negative: 98.4% of MW tests result were significant at $p < .05$, 89.9% at $p < .01$, and 61.1% at $p < .001$. No correlation was present between Billboard’s song ranking and a song’s LNS ($r = -.148$, $p = .140$).

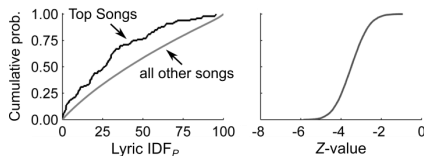


Figure 7. (a.) ECDFs of LNSs for the 100 Top Songs and the remaining 270,582 songs in the corpus. (b.) ECDF of Z -values from the 10,000 MW tests.

8.2 Billboard Top Artists analysis

Figure 8a shows the ECDFs of artist-level LNS for the set of 98 Top Artists and the remaining 5786 artists. As with the Top Songs, LNSs for the Top Artists are pulled towards zero, indicating reduced lexical novelty (i.e., lower IDF_p trimean values) for the set of 98 Top Artists. The Z -value distribution (Figure 8b) is more negative than in the Top Songs analysis: 99.3% of tests were significant at $p < .001$, 95.8% at $p < .0001$, and 85.5% at $p < .00001$. As with the Top Songs, no correlation was present between Billboard’s artist ranking and artist-level LNS ($r = -.059$, $p = .564$).

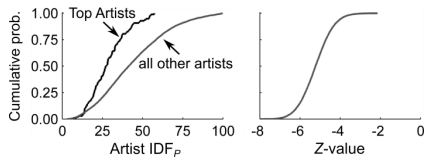


Figure 8. (a.) ECDFs of artist-level LNSs for the 98 Top Artists and the remaining 5786 artists in the corpus. (b.) ECDF of Z -values from the 10,000 MW tests.

¹⁴ billboard.com/articles/list/2155531/the-hot-100-all-time-top-songs

¹⁵ billboard.com/articles/columns/chart-beat/5557800/hot-100-55th-anniversary-by-the-numbers-top-100-artists-most-no

9. DISCUSSION

9.1 Summary

Stimulus novelty has influence over perception, memory, and affective response. Here, we define a *lexical novelty score* (LNS) for song lyrics. The LNS is derived from the inverse document frequency of all unique words in a lyric, and is scaled with respect to the number of unique words. Higher-order scores can be easily defined at the level of artists, albums, or genres, creating additional features for filtering operations or similarity assessments.

Although the construct validity of the LNS must be assessed by future user studies (see Section 9.2), a first-pass validation was performed by comparing LNSs associated with Billboard Magazine’s “official” lists of the 100 Top Songs and 100 Top Artists with LNSs from random sets of songs and artists. Lexical novelty was significantly lower—in a highly consistent way—for items on the Billboard lists, supporting the broad hypothesis that moderate stimulus novelty is preferred over high stimulus novelty [10–12].

The absence of any significant correlation between Billboard’s actual *ranking* of items on the Top Songs or Top Artists lists and our lexical novelty score should not be read as a “strike” against either Billboard’s methodology or our own. Rather, we regarded these lists as a source of well-known independent data that enabled us to make *a priori* predictions concerning differences in lexical novelty at the set (rather than the item) level.

6.2 Future directions

The present analyses of Billboard’s “Top 100” lists are but one of many analyses that could be performed. Further work could explore differences in lexical novelty among genres, subgenres, or styles (using external sources of metadata, such as Echo Nest¹⁶, Rovi¹⁷ or 7digital¹⁸); changes in lexical novelty over time (e.g., using lyric copyright date information); or correlations between lexical novelty and other performance-related metrics, such as RIAA-tracked album sales¹⁹.

A potential refinement of our LNS calculation would be to make it sensitive to parts of speech. Numerous English words can serve as multiple parts of speech, often with very different word frequencies. Capturing these usage patterns would, in principle, increase the sensitivity of the LNS. A revised SUBTLEX_{US} table of document frequencies is available that tallies parts-of-speech [42], as are widely used parts-of-speech taggers^{20,21}, making this modification tractable.

Finally, user studies must be performed to answer whether the proposed LNS *itself* has construct validity. These studies should evaluate, for example, whether lyrics with a high LNS yield longer reaction times and increased effort during a sentence processing task (e.g., as in [43]); or whether lyrics with a moderate LNS receive higher ratings of pleasure or liking than lyrics with either a low or a high LNS.

Together, these future steps will enhance the utility of the LNS in the context of music retrieval and recommendation applications.

10. DATA SET AVAILABILITY

With gratitude to LyricFind, much of the data presented here—lyrics in bag-of-words format; lyric, artist, and album IDs; and lyric- and artist-level lexical novelty scores—is made publically available for the first time: www.smcnus.org/lyrics/.

11. ACKNOWLEDGEMENT

Kind thanks to Roy Hennig, Director of Sales at LyricFind, for making this collaboration possible. This project was funded by the National Research Foundation (NRF) and managed through the multi-agency Interactive & Digital Media Programme Office (IDMPO) hosted by the Media Development Authority of Singapore (MDA) under Centre of Social Media Innovations for Communities (COSMIC).

12. REFERECNES

- [1] F. Kleedorfer, P. Knees, and T. Pohle, “Oh Oh Oh Whoah! Towards Automatic Topic Detection In Song Lyrics,” in *Proc. Int. Symp. Music Inf. Retrieval*, 2008, pp. 287–292.
- [2] R. Mayer, R. Neumayer, and A. Rauber, “Rhyme and Style Features for Musical Genre Classification by Song Lyrics,” in *Proc. Int. Symp. Music Inf. Retrieval*, 2008, pp. 337–342.
- [3] C. Laurier, J. Grivolla, and P. Herrera, “Multimodal music mood classification using audio and lyrics,” in *Proc. 7th Int. Conf. Mach. Learn. Appl.*, 2008, pp. 688–693.
- [4] X. Hu, J. S. Downie, and A. F. Ehmann, “Lyric text mining in music mood classification,” *Am. Music*, vol. 183, no. 5,049, pp. 2–209, 2009.
- [5] M. Van Zaanen and P. Kanter, “Automatic Mood Classification Using TF*IDF Based on Lyrics,” in *Proc. Int. Symp. Music Inf. Retrieval*, 2010, pp. 75–80.
- [6] X. Wang, X. Chen, D. Yang, and Y. Wu, “Music Emotion Classification of Chinese Songs based on Lyrics Using TF*IDF and Rhyme,” in *Proc. Int. Symp. Music Inf. Retrieval*, 2011, pp. 765–770.

¹⁶ <http://developer.echonest.com/docs/v4>

¹⁷ <http://developer.rovicorp.com>

¹⁸ <http://developer.7digital.com/>

¹⁹ <https://www.riaa.com/goldandplatinumdata.php>

²⁰ <http://ucrel.lancs.ac.uk/claws/trial.html>

²¹ <http://nlp.stanford.edu/software/lex-parser.shtml>

- [7] K. Rayner and S. A. Duffy, "Lexical complexity and fixation times in reading," *Mem. Cognit.*, vol. 14, no. 3, pp. 191–201, 1986.
- [8] F. Meunier and J. Segui, "Frequency effects in auditory word recognition," *J. Mem. Lang.*, vol. 41, no. 3, pp. 327–344, 1999.
- [9] M. Brysbaert and B. New, "Moving beyond Kučera and Francis," *Behav. Res. Methods*, vol. 41, no. 4, pp. 977–990, 2009.
- [10] D. E. Berlyne, *Aesthetics and Psychobiology*. New York: Appleton-Century-Crofts, 1971.
- [11] W. Sluckin, A. M. Colman, and D. J. Hargreaves, "Liking words as a function of the experienced frequency of their occurrence," *Br. J. Psychol.*, vol. 71, no. 1, pp. 163–169, 1980.
- [12] A. C. North and D. J. Hargreaves, "Subjective complexity, familiarity, and liking for popular music," *Psychomusicology*, vol. 14, no. 1, pp. 77–93, 1995.
- [13] M. Zuckerman, *Behavioral expressions and biosocial bases of sensation seeking*. Cambridge university press, 1994.
- [14] M. Kaminskas and F. Ricci, "Contextual music information retrieval and recommendation," *Comput. Sci. Rev.*, vol. 6, no. 2, pp. 89–119, 2012.
- [15] X. Wang, D. Rosenblum, and Y. Wang, "Context-aware mobile music recommendation for daily activities," in *Proc. 20th ACM Int. Conf. Multimedia*, 2012, pp. 99–108.
- [16] C. F. Mora, "Foreign language acquisition and melody singing," *ELT J.*, vol. 54, no. 2, pp. 146–152, 2000.
- [17] K. R. Paquette and S. A. Rieg, "Using music to support the literacy development of young English language learners," *Early Child. Educ. J.*, vol. 36, no. 3, pp. 227–232, 2008.
- [18] C. Y. Wan and G. Schlaug, "Music making as a tool for promoting brain plasticity across the life span," *The Neuroscientist*, vol. 16, no. 5, pp. 566–577, 2010.
- [19] G. R. Klare, "The measurement of readability," *ACM J. Comput. Doc. JCD*, vol. 24, no. 3, pp. 107–121, 2000.
- [20] T. G. Gunning, "The role of readability in today's classrooms," *Top. Lang. Disord.*, vol. 23, no. 3, pp. 175–189, 2003.
- [21] G. K. Berland, M. N. Elliott, L. S. Morales, J. I. Algazy, R. L. Kravitz, M. S. Broder, and others, "Health information on the Internet: accessibility, quality, and readability in English and Spanish," *J. Am. Med. Assoc.*, vol. 285, no. 20, pp. 2612–2621, 2001.
- [22] J. S. Chall and E. Dale, *Readability revisited: The new Dale-Chall readability formula*. Brookline Books, 1995.
- [23] G. Spache, "A new readability formula for primary-grade reading materials," *Elem. Sch. J.*, pp. 410–413, 1953.
- [24] M. Milone, "Development of the ATOS readability formula." Renaissance Learning, 2014.
- [25] G. Salton, A. Wong, and C.-S. Yang, "A vector space model for automatic indexing," *Commun. ACM*, vol. 18, no. 11, pp. 613–620, 1975.
- [26] A. Aizawa, "An information-theoretic perspective of tf-idf measures," *Inf. Process. Manag.*, vol. 39, no. 1, pp. 45–65, 2003.
- [27] J. Ramos, "Using tf-idf to determine word relevance in document queries," in *Proc. 1st Int. Conf. Machine Learning*, 2003.
- [28] S. Cronen-Townsend, Y. Zhou, and W. B. Croft, "Predicting query performance," in *Proc. 25th ACM SIGIR*, 2002, pp. 299–306.
- [29] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "Lp-norm idf for large scale image search," in *IEEE CVPR*, 2013, pp. 1626–1633.
- [30] J. Allan, C. Wade, and Alvaro Bolivar, "Retrieval and novelty detection at the sentence level," in *Proc. 26th ACM SIGIR*, 2003, pp. 314–321.
- [31] J. Foote, "Automatic audio segmentation using a measure of audio novelty," in *IEEE Int. Conf. Multimedia Expo.*, 2000, vol. 1, pp. 452–455.
- [32] T. McEnery and A. Hardie, *Corpus linguistics: Method, theory and practice*. Cambridge University Press, 2011.
- [33] T. Bertin-Mahieux, D. P. Ellis, B. Whitman, and P. Lamere, "The million song dataset," in *Proc. 12th Int. Symp. Music Inf. Retrieval*, 2011, pp. 591–596.
- [34] A. Caramazza, A. Laudanna, and C. Romani, "Lexical access and inflectional morphology," *Cognition*, vol. 28, no. 3, pp. 297–332, 1988.
- [35] G. Yu, "Lexical diversity in writing and speaking task performances," *Appl. Linguist.*, vol. 31, no. 2, pp. 236–259, 2010.
- [36] A. Xanthos, S. Laaha, S. Gillis, U. Stephany, A. Aksu-Koç, A. Christofidou, and others, "On the role of morphological richness in the early development of noun and verb inflection," *First Lang.*, p. 0142723711409976, 2011.
- [37] J. W. Tukey, *Exploratory data analysis*. Reading, MA: Addison-Wesley, 1977.
- [38] M. E. Newman, "Power laws, Pareto distributions and Zipf's law," *Contemp. Phys.*, vol. 46, no. 5, pp. 323–351, 2005.
- [39] E. T. Bradlow and P. S. Fader, "A Bayesian lifetime model for the 'Hot 100' Billboard songs," *J. Am. Stat. Assoc.*, vol. 96, no. 454, pp. 368–381, 2001.
- [40] D. E. Giles, "Survival of the hippest: life at the top of the hot 100," *Appl. Econ.*, vol. 39, no. 15, pp. 1877–1887, 2007.
- [41] R. M. Royall, "The effect of sample size on the meaning of significance tests," *Am. Stat.*, vol. 40, no. 4, pp. 313–315, 1986.
- [42] M. Brysbaert, B. New, and E. Keuleers, "Adding part-of-speech information to the SUBTLEX-US word frequencies," *Behav. Res. Methods*, vol. 44, no. 4, pp. 991–997, 2012.
- [43] A. D. Friederici, "Towards a neural basis of auditory sentence processing," *Trends Cogn. Sci.*, vol. 6, no. 2, pp. 78–84, 2002.

AN EFFICIENT TEMPORALLY-CONSTRAINED PROBABILISTIC MODEL FOR MULTIPLE-INSTRUMENT MUSIC TRANSCRIPTION

Emmanouil Benetos

Centre for Digital Music
Queen Mary University of London
emmanouil.benetos@qmul.ac.uk

Tillman Weyde

Department of Computer Science
City University London
t.e.weyde@city.ac.uk

ABSTRACT

In this paper, an efficient, general-purpose model for multiple instrument polyphonic music transcription is proposed. The model is based on probabilistic latent component analysis and supports the use of *sound state* spectral templates, which represent the temporal evolution of each note (e.g. attack, sustain, decay). As input, a variable-Q transform (VQT) time-frequency representation is used. Computational efficiency is achieved by supporting the use of pre-extracted and pre-shifted sound state templates. Two variants are presented: without temporal constraints and with hidden Markov model-based constraints controlling the appearance of sound states. Experiments are performed on benchmark transcription datasets: MAPS, TRIOS, MIREX multiF0, and Bach10; results on multi-pitch detection and instrument assignment show that the proposed models outperform the state-of-the-art for multiple-instrument transcription and is more than 20 times faster compared to a previous sound state-based model. We finally show that a VQT representation can lead to improved multi-pitch detection performance compared with constant-Q representations.

1. INTRODUCTION

Automatic music transcription is defined as the process of converting an acoustic music signal into some form of musical notation [16] and is considered a fundamental problem in the fields of music information retrieval and music signal processing. The core problem of automatic music transcription is multi-pitch detection (i.e. the detection of multiple concurrent pitches), which despite recent advances is still considered an open problem, especially for a large polyphony level and multiple instruments.

A large subset of music transcription approaches use *spectrogram factorization* methods such as non-negative matrix factorization (NMF) and probabilistic latent component analysis (PLCA), which decompose an input time-frequency representation into a series of note templates

and note activations. Several variants of the above methods propose more complex formulations compared to the original NMF/PLCA models, and also add musically- and acoustically-meaningful constraints. Such spectrogram factorization methods include amongst others [4, 8, 10, 13, 15, 18, 24]. Issues related to spectrogram factorization methods include: the choice of an input time-frequency representation, the ability to recognize instruments, the support of tunings beyond twelve-tone equal temperament, the presence or absence of a pre-extracted dictionary, the incorporation of any constraints, as well as computational efficiency (given ever-expanding collections and archives of music recordings).

In this paper, a model for multiple-instrument transcription is proposed, which uses a 5-dimensional dictionary of *sound state* spectral templates (sound states correspond to the various states in the evolution of a note, such as the attack, sustain, and decay states). The proposed model is based on PLCA and decomposes an input time frequency representation (in this case, a variable-Q transform spectrogram) into a series of probability distributions for pitch, instrument, tuning, and sound state activations. This model is inspired by a convolutive model presented in [4] that used a 4-dimensional dictionary and was able to transcribe a recording at $60 \times$ real-time. This model uses pre-shifted spectral templates across log-frequency, thus introducing a new dimension in the dictionary and eliminating the need for convolutions. Thus, tuning deviations from equal temperament are supported and at the same time this model only uses linear operations that result in a system that is more than 20 times faster compared to the system of [4]. In addition, temporal constraints using pitch-wise hidden Markov models (HMMs) are incorporated, in order to model the evolution of a note as a sequence of sound states. Experiments are performed on several transcription datasets (MAPS, MIREX multiF0, Bach10, TRIOS) and experimental results for the multi-instrument datasets using the proposed system outperform the state-of-the-art. Finally, we show that a VQT representation leads to an improvement in transcription performance compared to the more common constant-Q transform (CQT) representation, especially on the detection of lower pitches. Code for the proposed model is also supplied (cf. Section 4).

The outline of this paper is as follows. The proposed system is presented in Section 2. The employed training and test datasets, evaluation metrics, and experimental re-



© Emmanouil Benetos, Tillman Weyde.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Emmanouil Benetos, Tillman Weyde. "An efficient temporally-constrained probabilistic model for multiple-instrument music transcription", 16th International Society for Music Information Retrieval Conference, 2015.

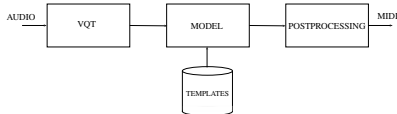


Figure 1. Diagram for the proposed system.

sults are shown in Section 3. Finally, a discussion on the proposed system followed by future directions is made in Section 4.

2. PROPOSED SYSTEM

2.1 Motivation

The overall aim of the proposed work is the creation of a system for automatic transcription of polyphonic music, that supports the identification of instruments along with multiple pitches, supports tunings beyond twelve-tone equal temperament along with frequency modulations, is able to model the evolution of each note (as a temporal succession of *sound states*), and is finally computationally efficient. The proposed system is based on work carried out in [4], which relied on a convolutive PLCA-based model and a 4-dimensional sound state dictionary. The aforementioned model was able to transcribe recordings at approximately $60 \times$ real-time (i.e. for a 1min recording, transcription took 60min). This paper proposes an alternative linear model able to overcome the computational bottleneck of using a convolutive model, which is supported by the use of a 5-dimensional dictionary of pre-extracted and pre-shifted sound state spectral templates, at the same time providing the same benefits with the model of [4]. Finally, this paper proposes the use of a variable-Q transform (VQT) representation, in contrast with the more common constant-Q transform (CQT) or linear frequency representations (a detailed comparison is made in Section 3). On related work, a linear model that used a 4-dimensional dictionary which did not support sound state templates or temporal constraints was proposed in [3].

In Fig. 1, a diagram for the proposed system can be seen. As motivation on the use of sound state templates, two log-frequency representations for a G1 piano note are shown in Fig. 2: it is clear that the note evolves from an attack/transient state to a steady state, and finally to a decay state. Fig. 3 shows 3 spectral templates extracted for the same note, which correspond to the 3 sound states (the lower corresponds to the attack state, the middle to the steady state and the top to the decay state).

2.2 PLCA-based model

The first variant of the proposed system takes as input a normalised log-frequency spectrogram $V_{\omega,t}$ (ω is the log-frequency index and t is the time index) and approximates it as a bivariate probability distribution $P(\omega,t)$. In this work, $V_{\omega,t}$ is a variable-Q time-frequency representation with a resolution of 60 bins/octave and minimum frequency

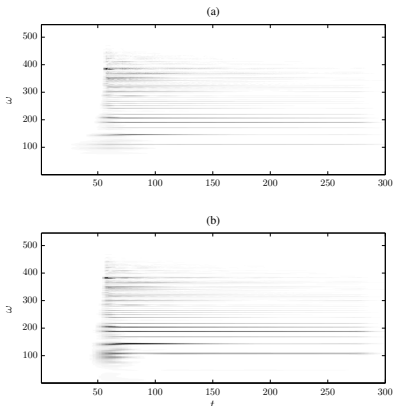


Figure 2. (a) The CQT spectrogram of a G1 piano note. (b) The VQT spectrogram for the same note.

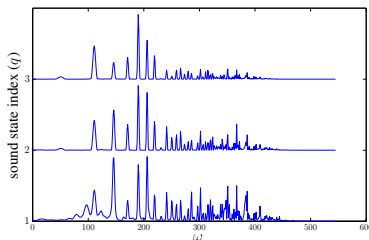


Figure 3. Sound state spectral templates for a G1 piano note (extracted using a VQT representation).

of 27.5Hz, computed using the method of [22]. As discussed in [22], a variable-Q representation offers increased temporal resolution in lower frequencies compared with a constant-Q representation. At the same time, a log-frequency transform represents pitch in a linear scale (where inter-harmonic spacings are constant for all pitches), thus allowing for pitch changes to be represented by shifts across the log-frequency axis.

In the model, $P(\omega,t)$ is decomposed into a series of log-frequency spectral templates per sound state, pitch, instrument, and log-frequency shifting (which indicates deviation with respect to equally tempered tuning), as well as probability distributions for sound state, pitch, instrument, and tuning activations. As explained in [4], a sound state represents different segments in the temporal evolution of a note; e.g. for a piano, different sound states can correspond to the attack, sustain, and decay.

The model is formulated as:

$$P(\omega, t) = P(t) \sum_{q,p,f,s} P(\omega|q,p,f,s) P_t(f|p) P_t(s|p) P_t(p) P_t(q|p) \quad (1)$$

where q denotes the sound state, p denotes pitch, s denotes instrument source, and f denotes log-frequency shifting. $P(t)$ is the energy of the log-spectrogram, which is a known quantity. $P(\omega|q,p,f,s)$ is a 5-dimensional tensor that represents the pre-extracted log-spectral templates per sound state q , pitch p and instrument s , which are also pre-shifted across log-frequency f . The proposed pre-shifting operation is made in order to account for pitch deviations, without needing to formulate a convolutive model across log-frequency, as in [4]. $P_t(f|p)$ is the time-varying log-frequency shifting distribution per pitch, $P_t(s|p)$ is the instrument source contribution per pitch over time, $P_t(q|p)$ is the time-varying sound state activation per pitch, and finally $P_t(p)$ is the pitch activation, which is essentially the resulting multi-pitch detection output.

In the proposed model, $f \in [1, \dots, 5]$, where $f = 3$ is the ideal tuning position for the template (using equal temperament). Given that the input time-frequency representation has a resolution of 5 bins per semitone, this means that all templates are pre-shifted across log-frequency on a ± 20 and ± 40 cent range around the ideal tuning position, thus accounting for small tuning deviations or frequency modulations. The proposed model also uses 3 sound states per pitch; more information on the extraction of the sound state spectral templates is given in subsection 3.1.

The unknown model parameters ($P_t(f|p)$, $P_t(s|p)$, $P_t(p)$, $P_t(q|p)$) can be iteratively estimated using the expectation-maximization (EM) algorithm [9]. For the *Expectation* step, the following posterior is computed:

$$P_t(q,p,f,s|\omega) = \frac{P(\omega|q,p,f,s) P_t(f|p) P_t(s|p) P_t(p) P_t(q|p)}{\sum_{q,p,f,s} P(\omega|q,p,f,s) P_t(f|p) P_t(s|p) P_t(p) P_t(q|p)} \quad (2)$$

For the *Maximization* step, unknown model parameters are updated using the posterior from (2):

$$P_t(f|p) = \frac{\sum_{\omega,s,q} P_t(q,p,f,s|\omega) V_{\omega,t}}{\sum_{f,\omega,s,q} P_t(q,p,f,s|\omega) V_{\omega,t}} \quad (3)$$

$$P_t(s|p) = \frac{\sum_{\omega,f,q} P_t(q,p,f,s|\omega) V_{\omega,t}}{\sum_{s,\omega,f,q} P_t(q,p,f,s|\omega) V_{\omega,t}} \quad (4)$$

$$P_t(p) = \frac{\sum_{\omega,f,s,q} P_t(q,p,f,s|\omega) V_{\omega,t}}{\sum_{p,\omega,f,s,q} P_t(q,p,f,s|\omega) V_{\omega,t}} \quad (5)$$

$$P_t(q|p) = \frac{\sum_{\omega,f,s} P_t(q,p,f,s|\omega) V_{\omega,t}}{\sum_{q,\omega,f,s} P_t(q,p,f,s|\omega) V_{\omega,t}} \quad (6)$$

Eqs. (2)-(6) are iterated until convergence; typically 15-20 iterations are sufficient. No update rule for the sound state templates $P(\omega|q,p,f,s)$ is included, since they are

considered fixed in the model. As in [4], we also incorporated sparsity constraints on $P_t(p)$ and $P_t(s|p)$ in order to control the polyphony level and the instrument contribution in the resulting transcription. The resulting multi-pitch detection output is given by $P(p,t) = P(t) P_t(p)$, while a time-pitch representation $P(f',t)$ can also be derived from the model, as in [4] (this representation has the same pitch resolution as in the input representation, i.e. 20 cent resolution).

2.3 Temporally-constrained model

This model variant proposes a formulation that expresses the evolution of each note as a succession of sound states, following work carried out in [4]. These temporal constraints are modelled using pitch-wise hidden Markov models (HMMs). This also follows the work done by Mysore in [17] on the non-negative HMM (a spectrogram factorization framework where the appearance of each template is controlled by an HMM).

As discussed, one HMM is created per pitch p , which has as hidden states the sound states q (assuming 88 pitches that cover the entire note range of a piano, 88 HMMs are used). Thus, the basic elements of this pitch-wise HMM are: the sound state priors $P(q_1^{(p)})$, the sound state transitions $P(q_{t+1}^{(p)}|q_t^{(p)})$, and the observations $P(\bar{\omega}_t|q_t^{(p)})$. Following the notation of [17], $\bar{\omega}$ corresponds to the sequence of observed spectra from all time frames, and $\bar{\omega}_t$ is the observed spectrum at the t -th time frame. Also, $q_t^{(p)}$ is the value of the hidden sound state at the t -th frame for pitch p .

In this paper, the model formulation is the same as in (1), where the following assumption is made:

$$P_t(q|p=i) = P_t(q_t^{(p=i)}|\bar{\omega}) \quad (7)$$

which means that the sound state activations are assumed to be produced by the posteriors (also called *responsibilities*) of the HMM for pitch p . Following [17], the observation probability is calculated as:

$$P(\bar{\omega}_t|q_t^{(p)}) = \prod_{\omega_t} P(\omega_t|q_t^{(p)}) V_{\omega,t} \quad (8)$$

where $P(\omega_t|q_t^{(p)})$ is the approximated spectrum for a given sound state and pitch. The observation probability is calculated as above since in PLCA-based models, $V_{\omega,t}$ represents the number of times ω has been drawn at the t -th time frame [17].

In order to estimate the unknown parameters of this proposed temporally-constrained model, the EM algorithm is also used, which results in a series of iterative update rules that combine PLCA-based updates as well as the HMM forward-backward algorithm [20]. For the Expectation step, the HMM posterior per pitch is computed as:

$$P_t(q_t^{(p)}|\bar{\omega}) = \frac{P_t(\bar{\omega}, q_t^{(p)})}{\sum_{q_t^{(p)}} P_t(\bar{\omega}, q_t^{(p)})} = \frac{\alpha_t(q_t^{(p)}) \beta_t(q_t^{(p)})}{\sum_{q_t^{(p)}} \alpha_t(q_t^{(p)}) \beta_t(q_t^{(p)})} \quad (9)$$

where $\alpha_t(q_t^{(p)})$ and $\beta_t(q_t^{(p)})$ are the forward and backward variables for the p -th HMM, respectively, and can be computed using the forward-backward algorithm [20]. The posterior for the transition probabilities $P_t(q_{t+1}^{(p)}, q_t^{(p)}|\bar{\omega})$ is also computed as in [4]. Finally, the model posterior is computed using (2) and (7).

For the Maximization step, unknown parameters $P_t(f|p)$, $P_t(s|p)$, and $P_t(p)$ are computed using eqs. (3)-(5). Finally, the sound state priors and transitions per pitch p are estimated as:

$$P(q_1^{(p)}) = P_1(q_1^{(p)}|\bar{\omega}) \quad (10)$$

$$P(q_{t+1}^{(p)}|q_t^{(p)}) = \frac{\sum_t P_t(q_t^{(p)}, q_{t+1}^{(p)}|\bar{\omega})}{\sum_{q_{t+1}^{(p)}} \sum_t P_t(q_t^{(p)}, q_{t+1}^{(p)}|\bar{\omega})} \quad (11)$$

In our experiments, it was found that an initial estimation of the pitch and source activations using the PLCA-only updates in the Maximization step leads to a good initial solution. In the final iterations (set to 3 in this case), the HMM parameters are estimated as well, which leads to an estimate of the sound state activations, and an improved solution over the non-temporally constrained model of subsection 2.2.

2.4 Post-processing

For both the non-temporally constrained model of subsection 2.2 and the temporally-constrained model of subsection 2.3, the resulting pitch activation $P(p, t) = P(t)P_t(p)$ (which is used for multi-pitch detection evaluation) as well as the pitch activation for a specific instrument $P(s, p, t) = P(t)P_t(p)P_t(s|p)$ (which is used for instrument assignment evaluation) need to be converted into a binary representation such as a piano-roll or a MIDI file. As in the vast majority of spectrogram factorization-based music transcription systems (e.g. [10, 15]), thresholding is performed on the pitch and instrument activations, followed by a process for removing note events with a duration less than 80ms.

3. EVALUATION

3.1 Training data

Sound state templates are extracted for several orchestral instruments, using isolated note samples from the RWC database [14]. Specifically, templates are extracted for bassoon, cello, clarinet, flute, guitar, harpsichord, oboe, piano, alto sax, and violin, using the variable-Q transform as a time-frequency representation [22]. The complete note range of the instruments (given available data) is used. The sound state templates are computed in an unsupervised manner, using a single-pitch and single-instrument variant of the model of (1), with the number of sound states set to 3.

3.2 Test data

Several benchmark and freely available transcription datasets are used for evaluation (all of them contain pitch ground truth). Firstly, thirty piano segments of 30s duration are used from the MAPS database using the ‘ENSTDkCl’ piano model. This test dataset has in the past been used for

System	\mathcal{F}	\mathcal{P}	\mathcal{R}
§2.2	70.08%	76.78%	65.27%
§2.3	71.56%	77.95%	66.89%

Table 1. Multi-pitch detection results for the MAPS-ENSTDkCl dataset using the proposed models.

multi-pitch evaluation (e.g. [7, 18], the latter also citing results using the method of [24]).

The second dataset consists of the woodwind quintet recording from the MIREX 2007 multiF0 development dataset [1]. The multi-track recording has been evaluated in the past either in its complete duration [4], or in shorter segments (e.g. [19, 24]).

Thirdly, we employ the Bach10 dataset [11], a multi-track collection of multiple-instrument polyphonic music, suitable for both multi-pitch detection and instrument assignment experiments. It consists of ten recordings of J.S. Bach chorales, performed by violin, clarinet, saxophone, and bassoon.

Finally, the TRIOS dataset [12] is also used, which includes five multi-track recordings of trio pieces of classical and jazz music. Instruments included in the dataset are: bassoon, cello, clarinet, horn, piano, saxophone, trumpet, viola, and violin.

3.3 Metrics

For assessing the performance of the proposed system in terms of multi-pitch detection we utilise the onset-based metric used in the MIREX note tracking evaluations [1]. A note event is assumed to be correct if its pitch corresponds to the ground truth pitch and its onset is within a ± 50 ms range of t ground truth onset. Using the above rule, precision (\mathcal{P}), recall (\mathcal{R}), and F-measure (\mathcal{F}) metrics can be defined:

$$\mathcal{P} = \frac{N_{tp}}{N_{sys}}, \quad \mathcal{R} = \frac{N_{tp}}{N_{ref}}, \quad \mathcal{F} = \frac{2 \cdot \mathcal{R} \cdot \mathcal{P}}{\mathcal{R} + \mathcal{P}} \quad (12)$$

where N_{tp} is the number of correctly detected pitches, N_{sys} is the number of detected pitches, and N_{ref} is the number of ground-truth pitches. For comparison with other state-of-the-art methods, we also use frame-based multiple-F0 estimation metrics, defined in [2], denoted as $\mathcal{P}_f, \mathcal{R}_f, \mathcal{F}_f$.

For the instrument assignment evaluations with the Bach-10 dataset, we use the pitch ground-truth of each instrument and compare it with the instrument-specific output of the system. As for the multi-pitch metrics, we define the following note-based instrument assignment metrics: $\mathcal{F}_v, \mathcal{F}_c, \mathcal{F}_s, \mathcal{F}_b$, corresponding to violin, clarinet, saxophone, and bassoon, respectively. We also use a mean instrument assignment metric, denoted as \mathcal{F}_{ins} .

3.4 Results

Experiments are performed using the two proposed model variants from Section 2: the non-temporally constrained version of subsection 2.2 and the HMM-constrained version of subsection 2.3. In both versions, the post-processing

System	\mathcal{F}	\mathcal{P}	\mathcal{R}
§2.2	71.75%	68.78%	74.98%
§2.3	72.50%	73.31%	71.71%

Table 2. Multi-pitch detection results for the MIREX multiF0 recording using the proposed models.

System	\mathcal{F}	\mathcal{P}	\mathcal{R}
§2.2	64.43%	56.99%	74.16%
§2.3	65.01%	57.35%	75.11%

Table 3. Multi-pitch detection results for the Bach10 dataset using the proposed models.

steps are the same. For the HMM-constrained model, the HMMs are initialized as ergodic, with uniform priors and state transition probabilities.

In terms of multi-pitch detection evaluation, results for the MAPS, MIREX, Bach10, and TRIOS datasets are shown in Tables 1, 2, 3, and 4, respectively. In all cases, the HMM-constrained model outperforms the non-temporally constrained model. The difference over the two models in terms of F-measure is more prominent for the MAPS dataset (1.48%) and the TRIOS dataset (1.81%) compared to the MIREX (0.75%) and Bach10 (0.58%) datasets. This can be attributed to the presence of piano in the MAPS and TRIOS datasets, compared to the woodwind/string instruments present in the other two datasets; since the piano is a pitched percussive instrument with a clear attack and transient state, the incorporation of temporal constraints on sound state evolution can be considered more important compared to bowed string and woodwind instruments, that do not exhibit a clear decay state. As an example of the transcription performance of the proposed system, Fig. 4 shows the resulting pitch activation for the MIREX multiF0 recording along with the corresponding ground truth.

Instrument assignment results for the Bach10 dataset are presented in Table 5. As can be seen, the performance of the proposed system regarding instrument assignment is much lower compared to multi-pitch detection, which this can be attributed to the fact that instrument assignment is a much more challenging problem, since it not only requires a correct identification of a note, but also a correct classification of that detected note to a specific instrument. It is worth noting however that a clear improvement is reported when using the temporally-constrained model over the model of subsection 2.2. That improvement is consistent across all instruments.

3.4.1 Comparison with state-of-the-art

On comparison of the proposed system with other state-of-the-art multi-pitch detection methods, for MAPS the proposed HMM-constrained method outperforms the spectrogram factorization transcription methods of [18] and [24] by 13.2% and 2.5% in terms of \mathcal{F} , respectively. It is however outperformed by the transcription system of [7] (4.9% difference); it should be noted that the system of [7] is

System	\mathcal{F}	\mathcal{P}	\mathcal{R}
§2.2	57.55%	64.60%	54.04%
§2.3	59.36%	60.18%	59.45%

Table 4. Multi-pitch detection results for the TRIOS dataset using the proposed models.

System	F_v	F_c	F_s	F_b	F_{rms}
§2.2	10.55%	39.99%	33.87%	40.80%	31.30%
§2.3	12.28%	41.55%	34.53%	42.33%	32.67%

Table 5. Instrument assignment results for the Bach10 dataset using the proposed models.

developed specifically for piano, in contrast with the proposed multiple-instrument system.

Regarding comparison on the MIREX recording, the proposed method outperforms the method of [6] by 3.9% in terms of \mathcal{F} . In terms of \mathcal{F}_f , the first 30sec of the MIREX recording were evaluated using the systems of [24] and [19], leading to $\mathcal{F}_f = 62.5\%$ and $\mathcal{F}_f = 59.6\%$, respectively. The proposed HMM-constrained method reaches $\mathcal{F}_f = 70.35\%$, thus outperforming the aforementioned systems.

For the Bach10 dataset, a comparison is made using the accuracy metric defined in [11]. The proposed HMM-constrained method reaches an accuracy of 72.0%, whereas the method of [11] reaches 69.7% (the latter results are with unknown polyphony level, for direct comparison with the proposed method).

Finally, for the TRIOS dataset, multi-pitch detection results were reported in [6], with $\mathcal{F} = 57.6\%$. The proposed method reaches for the HMM-constrained case $\mathcal{F} = 59.3\%$, thus outperforming the system of [6].

3.4.2 Comparing time-frequency representations

In order to evaluate the use of the proposed input VQT time-frequency representation, a comparative experiment is made using the proposed system and having as input a constant-Q representation (using the method of [21], with a 60 bins/octave log-frequency resolution as with the VQT). For the comparative experiments, the MAPS-ENSTDKCl dataset is employed and both the non-temporally constrained and HMM-constrained models are evaluated. The post-processing steps are exactly the same as in the proposed method. Results show that when using the constant-Q representation $\mathcal{F} = 63.98\%$ for the non-temporally constrained model and $\mathcal{F} = 65.51\%$ for the temporally-constrained model, which are both significantly lower when compared to using a VQT representation as input (cf. Table 1).

In order to show the improved detection performance of a VQT representation with respect to lower pitches, the transcription performance for the MAPS dataset was computed when only taking into account notes below or above MIDI pitch 60 (middle C in the piano). Using the VQT, $\mathcal{F} = 65.18\%$ for the lower pitches and $\mathcal{F} = 74.98\%$ for the higher pitches. In contrast when using the CQT,

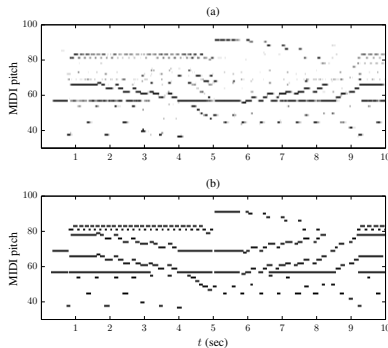


Figure 4. (a) The pitch activation output $P(p, t)$ for the first 10 sec of the MIREX multiF0 recording. (b) The corresponding pitch ground truth.

$\mathcal{F} = 51.17\%$ for the lower pitches and $\mathcal{F} = 74.58\%$ for the higher pitches. This result clearly demonstrates the benefit of using a VQT representation with respect to temporal resolution in lower frequencies, and by extension, to detecting lower pitches. As an example, Fig. 2 shows the CQT and VQT spectrograms for a G1 piano note, with the VQT exhibiting better temporal resolution in lower frequencies.

3.4.3 Sound state templates vs. note templates

Here, a comparison is performed between the use of the proposed 5-dimensional dictionary of sound state templates against the use of a 4-dimensional note template dictionary (which contains one template per pitch, instrument, and log-frequency shifting); the latter is supported by the method of [3]. In order to have a direct comparison, the method of [3] (for which the source code is publicly available) is modified as to use the same input VQT representation as well as post-processing steps with the proposed method, and is compared against the non-temporally constrained model of subsection 2.2.

When using a 4-dimensional dictionary, multi-pitch detection performance for the MAPS dataset reaches 64.65%, in contrast to 70.1% when using the 5-dimensional sound state dictionary. This shows the importance of using sound state templates, which are able to model the transient parts of the signal in contrast to simply using one (typically harmonic) note template for each pitch and instrument.

3.4.4 Runtimes

On computational efficiency, the proposed model requires linear operations like matrix/tensor multiplications in the EM steps; on the contrary, the previous model of [4] required the computation of convolutions which significantly slowed down computations. Regarding runtimes, the original HMM-constrained convolutive model of [4] runs at about $60 \times$ real-time using a Sony VAIO S15 laptop. Using the proposed method, the runtime is approximately 1

\times real-time for the non-temporally constrained model, and $2.5 \times$ real-time for the HMM-constrained model (i.e. for a 1min recording, runtimes are 1min and 2.5min, respectively). Thus, the proposed system is significantly faster compared to the model of [4], making it suitable for large-scale MIR applications.

4. CONCLUSIONS

In this paper, we proposed a computationally efficient system for multiple-instrument automatic music transcription, based on probabilistic latent component analysis. The proposed model employs a 5-dimensional dictionary of sound state templates, covering different pitches, instruments, and tunings. Two model variants were presented: a PLCA-only method and a temporally constrained model that uses pitch-wise HMMs in order to control the order of the sound states. Experiments were performed on several transcription datasets; results show that the temporally-constrained model outperforms the PLCA-based variant. In addition, the proposed system outperforms several state-of-the-art multiple-instrument transcription systems using the MIREX multiF0, Bach10, and TRIOS datasets. We also showed that a VQT representation can yield improved results compared to a CQT representation. Finally, the non-temporally constrained variant of the model is able to transcribe a recording at $1 \times$ real-time, thus making this method useful for large-scale applications. The Matlab code for the HMM-constrained model can be found online¹ in the hope that this model can serve as a framework for creating transcription systems useful to the MIR community.

This system can also be extended beyond the proposed formulations, by exploiting recent developments in spectrogram factorization-based approaches for music and audio signal analysis. Thus, the proposed model can also incorporate prior information in various forms (e.g. instrument identities, key information, music language models), following the PLCA-based approach of [23]. It can also use alternate EM update rules to guide convergence [8] or can use additional temporal continuity and sparsity constraints [13]. Drum transcription can also be incorporated into the system, in the same way as in [5]. In the future, we will also incorporate temporal constraints on note transitions and polyphony level estimation and will continue work on instrument assignment by combining timbral features with PLCA-based models.

5. ACKNOWLEDGEMENT

EB is supported by a Royal Academy of Engineering Research Fellowship (grant no. RF/128).

6. REFERENCES

- [1] Music Information Retrieval Evaluation eXchange (MIREX). <http://music-ir.org/mirexwiki/>.

¹https://code.soundsoftware.ac.uk/projects/amt_plca_5d

- [2] M. Bay, A. F. Ehmann, and J. S. Downie. Evaluation of multiple-F0 estimation and tracking systems. In *10th International Society for Music Information Retrieval Conference*, pages 315–320, Kobe, Japan, October 2009.
- [3] E. Benetos, S. Cherla, and T. Weyde. An efficient shift-invariant model for polyphonic music transcription. In *6th International Workshop on Machine Learning and Music*, Prague, Czech Republic, September 2013.
- [4] E. Benetos and S. Dixon. Multiple-instrument polyphonic music transcription using a temporally-constrained shift-invariant model. *Journal of the Acoustical Society of America*, 133(3):1727–1741, March 2013.
- [5] E. Benetos, S. Ewert, and T. Weyde. Automatic transcription of pitched and unpitched sounds from polyphonic music. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, pages 3131–3135, Florence, Italy, May 2014.
- [6] E. Benetos and T. Weyde. Explicit duration hidden Markov models for multiple-instrument polyphonic music transcription. In *14th International Society for Music Information Retrieval Conference*, pages 269–274, Curitiba, Brazil, November 2013.
- [7] T. Berg-Kirkpatrick, J. Andreas, and D. Klein. Unsupervised transcription of piano music. In *Advances in Neural Information Processing Systems*, pages 1538–1546, 2014.
- [8] T. Cheng, S. Dixon, and M. Mauch. A deterministic annealing em algorithm for automatic music transcription. In *14th International Society for Music Information Retrieval Conference*, Curitiba, Brazil.
- [9] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977.
- [10] A. Dessein, A. Cont, and G. Lemaitre. Real-time polyphonic music transcription with non-negative matrix factorization and beta-divergence. In *11th International Society for Music Information Retrieval Conference*, pages 489–494, Utrecht, Netherlands, August 2010.
- [11] Z. Duan, B. Pardo, and C. Zhang. Multiple fundamental frequency estimation by modeling spectral peaks and non-peak regions. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(8):2121–2133, November 2010.
- [12] J. Fritsch. High quality musical audio source separation. Master’s thesis, UPMC/IRCAM/ Télécom Paris-Tech, 2012.
- [13] B. Fuentes, R. Badeau, and G. Richard. Harmonic adaptive latent component analysis of audio and application to music transcription. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(9):1854–1866, September 2013.
- [14] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. RWC music database: music genre database and musical instrument sound database. In *International Conference on Music Information Retrieval*, Baltimore, USA, October 2003.
- [15] G. Grindlay and D. Ellis. Transcribing multi-instrument polyphonic music with hierarchical eigeninstruments. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1159–1169, October 2011.
- [16] A. Klapuri and M. Davy, editors. *Signal Processing Methods for Music Transcription*. Springer-Verlag, New York, 2006.
- [17] G. Mysore. *A non-negative framework for joint modeling of spectral structure and temporal dynamics in sound mixtures*. PhD thesis, Stanford University, USA, June 2010.
- [18] K. O’Hanlon and M.D. Plumbley. Polyphonic piano transcription using non-negative matrix factorisation with group sparsity. In *2014 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 3112–3116, May 2014.
- [19] P.H. Peeling and S.J. Godsill. Multiple pitch estimation using non-homogeneous poisson processes. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1133–1143, October 2011.
- [20] L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, February 1989.
- [21] C. Schörkhuber and A. Klapuri. Constant-Q transform toolbox for music processing. In *7th Sound and Music Computing Conf.*, Barcelona, Spain, July 2010.
- [22] C. Schörkhuber, A. Klapuri, N. Holighaus, and M. Dörfler. A Matlab toolbox for efficient perfect reconstruction time-frequency transforms with log-frequency resolution. In *AES 53rd Conference on Semantic Audio*, page 8 pages, London, UK, January 2014.
- [23] P. Smaragdis and G. Mysore. Separation by “humming”: user-guided sound extraction from monophonic mixtures. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, pages 69–72, New Paltz, USA, October 2009.
- [24] E. Vincent, N. Bertin, and R. Badeau. Adaptive harmonic spectral decomposition for multiple pitch estimation. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(3):528–537, March 2010.

ELECTRIC GUITAR PLAYING TECHNIQUE DETECTION IN REAL-WORLD RECORDINGS BASED ON F0 SEQUENCE PATTERN RECOGNITION

Yuan-Ping Chen, Li Su, Yi-Hsuan Yang

Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan

qoo0972@hotmail.com, lisu@citi.sinica.edu.tw, yang@citi.sinica.edu.tw

ABSTRACT

For a complete transcription of a guitar performance, the detection of playing techniques such as bend and vibrato is important, because playing techniques suggest how the melody is interpreted through the manipulation of the guitar strings. While existing work mostly focused on playing technique detection for individual single notes, this paper attempts to expand this endeavor to recordings of guitar solo tracks. Specifically, we treat the task as a time sequence pattern recognition problem, and develop a two-stage framework for detecting five fundamental playing techniques used by the electric guitar. Given an audio track, the first stage identifies prominent candidates by analyzing the extracted melody contour, and the second stage applies a pre-trained classifier to the candidates for playing technique detection using a set of timbre and pitch features. The effectiveness of the proposed framework is validated on a new dataset comprising of 42 electric guitar solo tracks without accompaniment, each of which covers 10 to 25 notes. The best average F-score achieves 74% in two-fold cross validation. Furthermore, we also evaluate the performance of the proposed framework for bend detection in five studio mixtures, to discuss how it can be applied in transcribing real-world electric guitar solos with accompaniment.

1. INTRODUCTION

Over the recent years there has been a flourishing number of online services such as Chordify¹ and Riffstation² that are dedicated to transcribing the chord progression of real-world guitar recordings [10]. As manual transcription demands on musical training and time, such services, despite not being perfect, make it much easier for music lovers and novice guitar learners to comprehend and appreciate

¹ <http://chordify.net/> (accessed: 2015-7-15)

² <http://play.riffstation.com/> (accessed: 2015-7-15)



© Yuan-Ping Chen, Li Su, Yi-Hsuan Yang.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Yuan-Ping Chen, Li Su, Yi-Hsuan Yang. "ELECTRIC GUITAR PLAYING TECHNIQUE DETECTION IN REAL-WORLD RECORDINGS BASED ON F0 SEQUENCE PATTERN RECOGNITION", 16th International Society for Music Information Retrieval Conference, 2015.

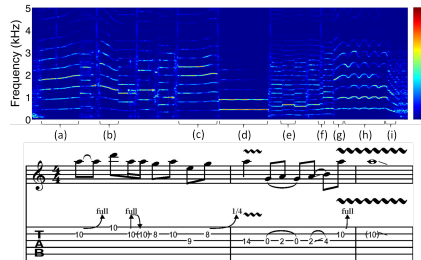


Figure 1. The spectrogram and tablature of a guitar phrase that contains the following techniques: bend (a, b, c, g), vibrato (d, h), hammer-on & pull-off (e) and slide (f, i).

music, thereby creating valuable educational, recreational and even cultural values.

For solo guitar recordings, a note-by-note transcription of the pitches and the playing techniques associated with each note is needed. While the sequence of notes constitutes a melody, playing techniques such as bend and vibrato determine how the notes are played and accordingly influence the expression of the guitar performance. As shown by the guitar tablature in Figure 1, a complete transcription of a guitar performance should contain the notations of the playing techniques.³

Unlike pitch estimation or chord recognition, research on playing technique detection is still in its early stages. Most of the existing work, if not all, is only concerned with audio recordings of pre-segmented individual single notes. For example, Abeßer *et al.* [1] collected around 4,300 bass guitar single notes to investigate audio based methods to distinguish between 10 bass guitar playing techniques. Reboursière *et al.* [20] evaluated a number of audio features to detect 6 playing techniques over 1,416 samples of hexaphonic guitar single notes. More recently, Su *et al.* [18] recorded 11,928 electric guitar single notes and investigated features extracted from the cepstrum and phase derivatives to detect 7 playing techniques. It is,

³ Fretted stringed instruments such as the guitar usually employ the tablature as the form of musical notation. Various arrows and symbols are used in a guitar tablature to denote the playing techniques. To "generate" the tablature from an audio recording, one would also need to predict the finger positions on the guitar fret, which is beyond the scope of this paper.

however, not clear how these methods can be applied to detect playing techniques in a real-world guitar solo track, such as the one shown in Figure 1.

The only exception, to our best understanding, is the work presented by Kehling *et al.* [16], which extended the work presented in [1] and considered playing technique detection in 12 phrases of guitar solo. They proposed to use onset and off detection first to identify each note event in a guitar solo track, after which the statistics (*e.g.* minimum, maximum, mean, or median) of frame-level spectral features over the duration of each note event are extracted and fed to a pre-trained classifier for playing technique detection. Using the multi-class support vector machine (SVM) with radial basis function (RBF) kernel, they obtained 83% average accuracy in distinguishing between the following 6 cases: *normal*, *bend*, *slide*, *vibrato*, *harmonics*, and *dead notes*. It appears that lower recall rates are found for slide, vibrato, and bend: the recall rates are 50.9%, 66.7%, and 71.3%, respectively.

Although Kehling *et al.*'s work represented an important step forward in playing technique classification, their approach has a few limitations. First, using the whole note event as a fundamental unit in classification cannot deal with techniques that are concerned with the transition between successive notes, such as pull-off and hammer-on, which are also widely used in guitar. Second, extracting features from the whole note may include information not relevant to techniques that are related to only the beginning phase of note events, such as bend and slide. Third, existing techniques for onset and offset detection may not be robust to timbre variations commonly seen in guitar performance [2, 14], originating from the predominant use of sound effects such as distortion or delay [9]. Onset and offset detection would be even more challenging in the presence of accompaniments such as bass and drums.

In light of the above challenges, we propose in this work a new approach to playing technique detection in guitar, by exploiting the time sequence patterns over the melody contour. Given a guitar recording, our approach firstly employs a melody extraction algorithm to estimate the melody contour, *i.e.* sequence of fundamental frequency (F0) estimates. Then, we apply a number of signal processing methods to analyze the estimated melody contour, from which candidate regions of target playing techniques are identified. Because the candidates are identified from the melody contour, the proposed approach can deal with techniques employed during the transition or the beginning phase of notes. The candidate selection algorithms are designed in such a way that emphasizes more on recall rates. Finally, we further improve the precision rates by extracting spectral and pitch features from the candidate regions and using SVM for classification.

The effectiveness of the proposed approach is validated on a new dataset comprising of 42 electric guitar solos taken from the teaching material of the textbook, *Rock Lead Basics: Techniques, Scales and Fundamentals for Guitar*, by Danny Gill and Nick Nolan [13]. While the guitar phrases employed in Kehling *et al.*'s work are not

associated with any sound effect [16], the phrases we take from this book are recorded with distortion sound effect and are perceptually more melodic and realistic. Moreover, according to the data from the book, we consider the following five playing techniques in this work: *slide*, *vibrato*, *bend*, *hammer-on*, and *pull-off*, which are viewed as the most frequently used and fundamental techniques in rock lead guitar by the textbook authors.

The guitar solos collected from the book are not accompanied by any other instruments. To examine how the proposed approach can be applied to real-world recordings with accompaniment, we also conduct a preliminary evaluation using 5 well-known guitar solo tracks with different tones and accompaniments. The use of a source separation algorithm as a pre-processing step to suppress the accompaniments is also investigated.

2. DATASETS AND PLAYING TECHNIQUES

Two datasets are employed in this work. The first one is composed of 42 tracks of unaccompanied electric guitar solo obtained from the CD of the textbook by Danny Gill and Nick Nolan. The duration of the tracks is about 15–20 seconds, summing up to about 10 minutes. The tracks are recorded by a standard tuned electric guitar with clean tone and distortion sound effect, covering 10–25 notes per track. For evaluation purposes, we have the timestamps of the playing techniques employed in each track carefully annotated by an experienced electric guitar player, with the help of the corresponding guitar tablature. In total, we have 143 pull-offs, 70 hammer-ons, 143 bends, 74 slides, and 61 vibratos. While the audio tracks are copyright protected, we have made the annotations publicly available with the research community through a project webpage.⁴

The first dataset contains a variety of different possible realizations of the techniques in real-world performances. To illustrate this, we combine a few passages of different phrases and show in Figure 1 the spectrogram and guitar tab. The five playing techniques and their possible variations are described below.

- **Bend** refers to stretching the string with left hand to increase the pitch of the bended note either gradually or instantly. The region (a) in Figure 1 shows a note *full-bended* from A4 to E4 gradually. In (b), the note is *pre-bended* to E4, *i.e.* bend the note without sounding it, and then *released* to A4 with the hitting of string. Region (c) shows a *half-step* bend commonly seen in Blues. A *grace note* bend is when you strike the string and at the same time bend the note to the target, as shown in (g).
- **Vibrato** represents minute and rapid variations in pitch. Regions (d) and (h) of Figure 1 show a very subtle vibrato with smaller extent and a wide vibrato with larger extent, respectively.

⁴<http://mac.citi.sinica.edu.tw/GuitarTranscription>. Note that we label the instant of transition between two notes for pull-off and hammer-on, the middle of the employment of bend and slide, and the whole duration (including the beginning and end timestamps) for vibrato.

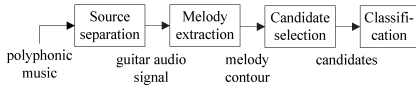


Figure 2. Flowchart of the proposed approach to guitar playing technique detection.

- **Hammer-on** is when a note is sounded, a left hand finger is used to quickly press down a fret that is on the same string while the first note is still ringing.
- **Pull-off** is when you have strummed one note and literally pull off of the string to a lower note. Rapid and successive use of pull-off and hammer-on is often referred to as *trill*, which is illustrated in (e).
- **Slide** refers to the action of slide left hand finger across one or more frets to reach another note. A slide between B3 and D4 is shown in (f). There are *shift* slides and *legato* slides. A guitar solo usually begins or ends with another variant known as *slide from/into nowhere*, which is illustrated in (i).

The second dataset, on the other hand, consists of 5 excerpts of real-world guitar solo (with accompaniment) clipped from the following famous recordings: segments 1'48"–2'39" and 2'51"–3'23" from *Bold as Love* by Jimi Hendrix, segments 0'17"–1'26" and 3'50"–4'33" from *Coming Back to Life* by Pink Floyd, and segment 4'22"–5'04" from *Wet Sand* by Red Hot Chili Peppers. The first two are both played in fuzzy tone (akin to overdrive), the third one with reverb effect in clean tone, the fourth one in overdrive, and the fifth one is played with the distortion effect. The excerpts last 3 minutes 57 seconds in total. We also manually label the playing techniques for evaluation.

3. PROPOSED APPROACH

3.1 Overview

Kehling *et al.* [16] employs a two-stage structure in detecting playing techniques in audio streams. The first stage uses onset and offset detection to identify each note event from the given audio track, and the second stage applies a pre-trained classifier to the note events for multiclass classification. A similar two-stage structure is also adopted in the proposed approach, but in our first stage we make use of the melody contour extracted from the given audio track, and employ a number of algorithms to identify candidates of playing techniques from the melody contour. Different candidate selection algorithms are specifically designed for the 5 playing techniques. Depending on the target playing technique, the input to the second-stage classifier can be temporal segments falling between note events or fragments of whole note events. In this way, the proposed approach can deal with techniques such as hammer-on and pull-off, while Kehling *et al.*'s approach cannot.

Figure 2 shows the flowchart of the proposed approach, which includes source separation as an optional pre-processing step to cope with instrumental accompaniments. We provide the details of each component below.

3.2 Source Separation

In real-world guitar performance, the guitar solo is usually mixed with strong bass line, percussion sounds, or others. Due to the accompaniments, the performance of estimating the melody contour of the lead guitar may degrade.

We experiment with the robust principal component analysis (RPCA) algorithm [6, 7, 15] to separate the sound of the lead guitar from the accompaniments, before extracting the melody. Given the magnitude spectrogram $\mathbf{D} \in \mathbb{R}^{t \times m}$ of the mixture, where t denotes temporal length and m the number of frequency bins, RPCA seeks to decompose \mathbf{D} into two matrices of the same size, a low-rank matrix \mathbf{A} and a sparse matrix \mathbf{E} , by solving the following convex optimization problem:

$$\min_{\mathbf{A}, \mathbf{E}: \mathbf{D} = \mathbf{A} + \mathbf{E}} \|\mathbf{A}\|_* + \lambda \|\mathbf{E}\|_1, \quad (1)$$

where the trace norm $\|\cdot\|_*$ and l_1 norm $\|\cdot\|_1$ are convex surrogate of the rank and the number of nonzero entries of a matrix, respectively [6], and λ is a positive weighting parameter. As the *background* component of a signal is usually composed of repetitive elements in time or frequency, its spectrogram is likely to have a lower rank comparing to that of the *foreground*. RPCA has been applied to isolating the singing voice (foreground) from the accompaniment (background) [15]. We use the same idea, assuming that the guitar solo is the foreground (i.e. in \mathbf{E}).

3.3 Melody Extraction

Melody extraction has been an active field of research in the music information retrieval society for years [5, 8, 19]. It is concerned with the F0 sequence of only the main melody line in a polyphonic music recording. Therefore, it consists of a series of operations for creating candidate pitch contours from the F0 estimates and for selecting one of the pitch contours as the main melody. We employ the state-of-the-art melody extraction algorithm proposed by Salamon and Gómez [21], for its efficiency and well-demonstrated effectiveness. Specifically, we employ the implementation of the MELODIA algorithm developed by the authors for an open-source library called Essentia [3]. It is easy to use and the estimate is in general accurate.

3.4 Candidate Selection (CS)

We propose to mine the melody contour for the following time sequence patterns specific to each playing technique. Following this process of pattern finding, we can find candidates of the playing techniques scattered in the time flow of a music signal. We refer to this process as candidate selection, or CS for short.

- Bend: arc-like or twisted trajectories.
- Vibrato: sinusoidal patterns.
- Slide: upward or downward stair-like patterns.
- Hammer-on, pull-off: two adjacent parallel horizontal lines resulting from two notes of different F0s.

Clearly, such patterns may not necessarily correspond to true instances (or, true positives) of the playing techniques. For example, sounding two notes with pick picking

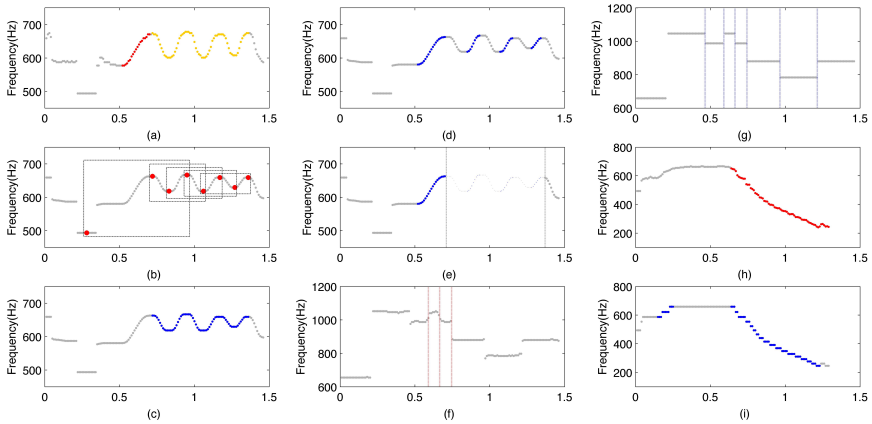


Figure 3. The procedure of candidate selection (best seen in color). (a) The raw melody contour of a bend (red segment) and a vibrato (yellow segment). (b) The processed melody contour by median filter, note tracking and mean filter. Four local extrema of pitch value create a window to determine vibrato. (c) The candidate segment for vibrato (blue). (d) The candidate segments for bend (blue). (e) The candidate segments for bend, after excluding candidates of vibrato (blue). (f) The raw melody contour of a pull-off and a hammer-on. The red vertical lines show the groundtruth instants of the playing techniques. (g) The processed melody contour by note tracking and quantization, and the blue vertical lines denote the candidates instants. (h) The raw melody contour of a “slide into nowhere” (red segment). (i) The processed melody contour by quantization, and the selected candidates for slide (blue segments).

also results in a pitch trajectory of two parallel horizontal lines akin to the case of hammer-on or pull-off. There might also be errors in the estimate of the melody contour (e.g. when the lead instrument is silent, the estimated melody contour may correspond to the sounds of other instruments). Therefore, the purpose of the CS process is actually to identify the candidates with high recall rates (i.e. not missing the true positives) and moderate precision rates (i.e. it is fine to have false positives). In the next stage, we will use SVM classifiers that are discriminatively trained to distinguish between true positives and false positives by exploiting both timbre and pitch features computed from these candidates. Because the CS process only considers pitch information, the additional use of timbre information in the classification stage has the potential to boost the precision rates.

As described below, the CS process is accomplished with a few simple signal processing methods for simplicity and efficiency. The methods are illustrated in Figure 3.

3.4.1 Vibrato and Bend

We use similar procedures to select the candidates of vibrato and bend, because the two techniques share the same arc-like trajectories. Indeed, a vibrato can be viewed as succession of bend up and then releasing down. The two techniques mainly differ in the number of the cycles. The following operations are firstly employed to process the (raw) melody contour estimated by MELODIA [3].

- First, we flatten the rugged raw contour and remove the outliers produced by the melody extraction algorithm by a 10 points (100ms in 44.1 kHz sampling rate) *median filter*, whose length is approximately shorter than a period of vibrato. The median filter also slightly corrects octave errors made by melody tracking.
- Second, we perform a simple *note tracking* step by grouping adjacent F0s into the same note if the pitch difference between them is smaller than 80 cents, according to the auditory streaming cues [4]. The step leads to a number of segments corresponding to different note events, from which segments shorter than 80ms are discarded, assuming that the a single note should last at least 80ms, approximately the length of a semiquaver in 180 BPM.
- Finally, we convolve each segment with a 5 points (50ms) *mean filter* with hop of 10ms for smoothing.

The segments are then considered as possible note events. We then use different ways to detect vibrato and bend. For vibrato, we search for all the local maxima and minima in each note [12]. A temporal fragment of four consecutive extrema within a note is considered as a vibrato candidate if the following conditions meet: 1) the temporal distance between two neighboring extrema should fall within 30ms and 400ms for valid vibrato rate, i.e., the modulation frequency from 1.25Hz to 16.67Hz; 2) the pitch difference between neighboring extrema should

be smaller than 225 cents, which is slightly larger than a whole note; 3) dividing the fragment into three shorter fragments of pitch sequence by the four extrema, the variance in the logarithmic pitch of each short fragment should be larger than an empirical threshold. Please see Figure 3(c) for an example.

On the other hand, we consider a temporal fragment as a bend candidate if the following conditions meet: 1) it is not a vibrato candidate; 2) the pitch sequence continuously ascends or descends for more than 80ms; 3) the pitch difference between two neighboring frames is smaller than 50 cents. An example can be found in Figure 3(e).

3.4.2 Pull-off and Hammer-on

While bend and vibrato can last a few frames, pull-off and hammer-on are considered as the temporal instance (*i.e.* a frame) during the transition of notes. Therefore, without using either a median or mean filter, we perform the note tracking procedure described in Section 3.4.1, and then *quantize* each F0 to its closest semitone in terms of cent. After this, we consider all the temporal instances in the middle of two notes as a candidate for both pull-off and hammer-on, as long as the following conditions meet: 1) the gap between the note transition is shorter than 20ms; 2) the pitch cannot be away from its closest semitone by 35 cents. The former condition is set, because it is known that the contact of pick (or right hand finger) and the string would temporarily stop the vibration of the string when a note is sounded by plucking the string, thereby creating the gap in the note transition [20]. The latter condition is set because there might be such gaps within the employment of a vibrato or a bend due to the F0 quantization.

Because each candidate for pull-off or hammer-on only lasts one frame, to characterize the temporal moment, we use a 100ms fragment centering at the candidate frame for the feature extraction step described in Section 3.5.

3.4.3 Slide

To recognizing the ladder-like pitch sequence pattern, we simply quantize all the F0s into its closest semitone without any pre-processing, in order not to falsely remove the transition notes of a bend (which is usually around tens of milliseconds). After quantization, we search for the ladders in the melody contour with the following rules: 1) the number of steps should be at least three (*i.e.* slide across at least three frets); 2) the length of transitional steps (notes) should fall within 10 to 70ms, according to our empirical observation from the data; 3) the pitch difference between neighboring steps should be exactly one semitone (*i.e.* a fret). Please refer to Figure 3(i) for an example.

3.5 Feature Extraction and Classification

After applying CS, we would have candidates of the 5 playing techniques spreading over the input guitar track. As we have mentioned, our design of the signal processing methods and the setting of some parameter values have been informed by the need of reaching high recall rate. It is then the job of the classifier to identify false positives of the

techniques and improve precision rates. The candidates are represented by the following three sets of audio features.

- **TIMBRE (T)** includes the statistics of the following features: spectral centroid, brightness, spread, skewness, kurtosis, flux, roll-off, entropy, irregularity, roughness, inharmonicity, zero-crossing rate, low-energy ratio, and their 1st-order time difference. We use the mean, standard deviation (STD), maximum, minimum, skewness, kurtosis as the statistics measure, so there are $13 \times 8 \times 2 = 208$ features in total.
- **MFCC (M)** contains mean and STD of the 40-D Mel-frequency cepstral coefficients and its 1st-order time difference, totalling 160 features. Both the TIMBRE and MFCC sets are computed by the open-source library MIRtoolbox [17].
- **Pitch (P)** is computed from the log scale F0 sequence on the processed (instead of the raw) melody contour. Except for vibrato, we use the following 6 features for all the playing techniques: skewness, kurtosis, variance, the difference between the maximum and minimum, and the mean and STD of the 1st-order time difference. For vibrato, as there are 3 short temporal fragments for each candidate (see Section 3.4.1), we calculate the 6 features for each of the fragment, and additionally use the variance of difference between the four pitch extrema in log scale and the variance of the temporal distance between the four pitch extrema, totalling 20 features.

4. EXPERIMENT

4.1 Experimental Setup

For short-time Fourier transform, we use the Hamming window of 46ms and 10ms overlap under the sampling rate of 44.1 kHz. For MELODIA, we set the lowest and highest possible F0 to 77Hz (E2b) and 1400 (F6) respectively, considering the frequency range of a standard-tuned guitar plus additionally half step tolerance of inaccurate tuning. We train 5 binary linear kernel SVMs [11], one for each technique,⁵ and employ z-score normalization for the features. The parameter C of SVM is optimized by an inside cross validation on the training data. We conduct training and testing 10 times under a two-fold cross validation scheme and report the average result, in terms of precision, recall and F-score. An estimate of bend or slide is deemed correct as long as the ground truth timestamp falls between the detected bend or slide segment. An estimate of pull-off or hammer-on is deemed correct if the detected instant of employment falls between the interval of ground truth instant with a tolerance time-window of 50ms. Vibrato is evaluated in the frame level, *e.g.* the recall of vibrato is the proportion of frames labeled as vibrato which are detected as vibrato. For evaluation on the studio mixtures, the SVM is trained over the 42 unaccompanied phrases. Source separation is only performed for the 5 studio mixtures.

⁵ It would have been better if a multi-class classification scheme is adopted to avoid possible overlaps of the estimates of different techniques. We leave the issue as a future work.

	Bend	Vibrato	Pull-off	Hammer-on	Slide
Recall	94.4	94.2	94.4	94.3	85.1
Precision	53.1	63.0	30.1	24.7	15.0
F-score	68.0	75.5	45.7	39.2	25.5

(a)

	Bend	Vibrato	Pull-off	Hammer-on	Slide
Recall	86.2	79.5	73.6	65.7	58.6
Precision	89.3	89.1	75.3	66.7	56.8
F-score	87.7	84.0	74.4	66.3	57.7

(b)

Table 1. Recall, precision, and F-scores (in %) of playing technique detection in the unaccompanied set using (a) CS only and (b) CS+SVM{MFCC,TIMBRE,Pitch}.

4.2 Experiment Result

4.2.1 Evaluation on Unaccompanied Guitar Solos

Table 1 shows the per-class result of playing technique detection over the 42 unaccompanied guitar solos, using either (a) only candidate selection (CS) or (b) both CS and SVM. The following observations can be made.

- Except for slide, the proposed CS methods can lead to recall rates higher than 94% for the considered playing techniques. Slide appears to be the most challenging one, as its detection can be affected by octave errors from the melody extraction step.
- By comparing Tables 1(a) and (b), we see that the second-stage SVM can remarkably improve the precision rates, and accordingly the F-scores, for all the playing techniques. This validates the effectiveness of the proposed approach.
- Bend and vibrato appear to be easier to detect, with F-scores 87.7% and 84.0%, respectively. Although it is not fair to compare the numbers with the ones reported in [16] due to differences in settings and datasets, the performance of the proposed approach seems to be promising. Interestingly, slide appears to be the most challenging case in our study and the one presented by [16], with comparable F-scores (57.7% versus 50.9%).

Figure 4 provides the F-scores of using different features in the SVM. Although not shown in the figure, MFCC appears to be the best performing individual feature set among the three. Better result is seen by concatenating the features (*i.e.* early fusion). Pitch features contribute more to the detection of hammer-on but less for others, possibly because pitch information has been exploited in CS.

4.2.2 Evaluation on Real-World Studio Mixtures

As bend detection is found to be promising, we focus on bend detection for our evaluation over the 5 studio mixtures, which include in total 85 bends. Figure 5 compares the F-score of bend detection of various methods, including the case when RPCA is employed before melody extraction. It is not surprising that the F-scores are lower than that obtained for the unaccompanied tracks. However, it is interesting to note that the best result can be obtained by CS only, regardless of whether RPCA or SVM is

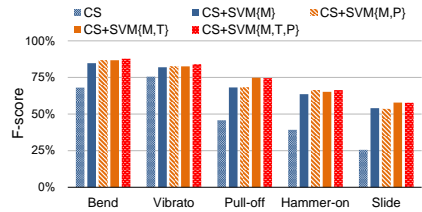


Figure 4. F-scores of playing technique detection in 42 unaccompanied guitar solos using various methods.

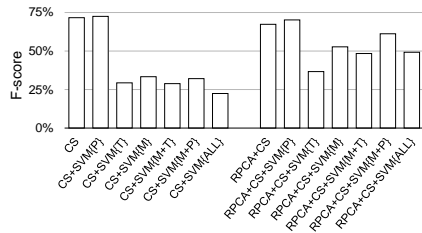


Figure 5. F-scores of bend detection of 5 accompanied guitar solos, without (left) or with (right) RPCA.

used. Actually, the result of using CS+SVM degrades a lot comparing to the case of CS only, except for the case that pitch features are considered in SVM. The performance of CS+SVM can be improved by using RPCA, but the result is still inferior to the result of CS only. We conjecture that the inferior result of CS+SVM can be attributed to the difference between the data used for training the SVM (*i.e.* the unaccompanied tracks) and the data for testing (*i.e.* the mixtures). The result might be better if we have a few training data that are with accompaniment. However, if such data are not available, it seems to be advisable to use the CS process only for the bend detection in mixtures.

5. CONCLUSION

In this paper, we have presented a two-stage approach for detecting 5 guitar playing techniques in guitar solos. The proposed approach is characteristic of its use of time sequence patterns mined from the melody contour of the lead guitar for candidate selection in the first stage, and then using classifiers to refine the result in the second stage. The F-scores for the unaccompanied set range from 57.7% to 87.7% depending on the playing techniques. The average F-score across the techniques reaches 74%. We have also evaluated the case of bend detection for a few guitar solos with accompaniment, and shown that the best F-score 67.3% is obtained by candidate selection alone.

6. ACKNOWLEDGMENT

This work was supported by the Academia Sinica Career Development Program.

7. REFERENCES

- [1] J. Abeßer, H. Lukashevich, and G. Schuller. Feature-based extraction of plucking and expression styles of the electric bass guitar. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pages 2290–2293, 2010.
- [2] S. Bock and G. Widmer. maximum filter vibrato suppression for onset detection. In *Proc. of the 16th Int. Conf. on Digital Audio Effects (DAFx)*, 2013.
- [3] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J. Zapata, and X. Serra. Essentia: an audio analysis library for music information retrieval. In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 493–498, 2013. [Online] <http://essentia.upf.edu>.
- [4] A. S. Bregman, editor. *Auditory scene analysis*. MIT Press, 1990.
- [5] P. M. Brossier. *Automatic Annotation of Musical Audio for Interactive Applications*. PhD thesis, Queen Mary, University of London, 2006.
- [6] E. J. Candès, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Journal of the ACM*, 58(3):1–37, 2011.
- [7] T.-S. Chan, T.-C. Yeh, Z.-C. Fan, H.-W. Chen, L. Su, Y.-H. Yang, and R. Jang. Vocal activity informed singing voice separation with the iKala dataset. In *Proc. IEEE Int. Conf. Acoust., Speech and Signal Process.*, pages 718–722, 2015.
- [8] A. De Cheveigné and H. Kawahara. Yin, a fundamental frequency estimator for speech and music. *The Journal of the Acoustical Society of America*, 111(4):1917–1930, 2002.
- [9] J. Dattorro. Effect design, part 2: Delay line modulation and chorus. *J. Audio engineering Society*, 45(10):764–788, 1997.
- [10] W. B. de Haas, J. P. Magalhães, and F. Wiering. Improving audio chord transcription by exploiting harmonic and metric knowledge. In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 295–300, 2012.
- [11] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: A library for large linear classification. *J. Machine Learning Research*, 2008. <http://www.csie.ntu.edu.tw/~cjlin/liblinear/>.
- [12] A. Friberg and E. Schoonderwaldt. Cuex: An algorithm for automatic extraction of expressive tone parameters in music performance from acoustic signals. *Acta Acustica united with Acustica*, 93(3):411–420, 2007.
- [13] D. Gill and N. Nolan. *Rock Lead Basics: Techniques, Scales and Fundamentals for Guitar*. Musicians Institute Press, Los Angeles, California, 1997.
- [14] A. Holzapfel, Y. Stylianou, A. C. Gedik, and B. Bozkurt. Three dimensions of pitched instrument onset detection. *IEEE Trans. Audio, Speech, and Language Processing*, pages 1517–1527, 2010.
- [15] P.-S. Huang, S. D. Chen, P. Smaragdis, and M. Hasegawa-Johnson. Singing-voice separation from monaural recordings using robust principal component analysis. In *Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing*, pages 57–60, 2012.
- [16] C. Kehling, J. Abeßer, C. Dittmar, and G. Schuller. Automatic tablature transcription of electric guitar recordings by estimation of score-and instrument-related parameters. In *Proc. Int. Conf. Digital Audio Effects*, 2014.
- [17] O. Lartillot and P. Toivainen. MIR in Matlab (II): A toolbox for musical feature extraction from audio. In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 127–130, 2007. [Online] <http://users.jyu.fi/~lartillo/mirttoolbox/>.
- [18] L. Su, L.-F. Yu, and Y.-H. Yang. Sparse cepstral and phase codes for guitar playing technique classification. In *Proc. Int. Soc. Music Information Retrieval Conf.*, pages 9–14, 2014.
- [19] M. Müller, D. P. W. Ellis, A. Klapuri, and G. Richard. Signal processing for music analysis. *IEEE J. Sel. Topics Signal Processing*, 5(6):1088–1110, 2011.
- [20] L. Reboursière, O. Lähdeoja, T. Drugman, S. Dupont, C. Picard, and N. Riche. Left and right-hand guitar playing techniques detection. In *Proc. Int. Conf. New Interfaces for Musical Expression*, 2012.
- [21] J. Salamon and E. Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Trans. Audio, Speech, and Language Processing*, 20(6):1759–1770, 2012.

EXTENDING A MODEL OF MONOPHONIC HIERARCHICAL MUSIC ANALYSIS TO HOMOPHONY

Phillip B. Kirlin and David L. Thomas

Department of Mathematics and Computer Science, Rhodes College

kirlinp@rhodes.edu, thodl15@gmail.com

ABSTRACT

Computers are now powerful enough and data sets large enough to enable completely data-driven studies of Schenkerian analysis, the most well-established variety of hierarchical music analysis. In particular, we now have probabilistic models that can be trained via machine learning algorithms to analyze music in a hierarchical fashion as a music theorist would. Most of these models, however, only analyze the monophonic melodic content of the music, as opposed to taking all of the musical voices into account. In this paper, we explore the feasibility of extending a probabilistic model developed for analyzing monophonic music to function with homophonic music. We present details of the new model, an algorithm for determining the most probable analysis of the music, and a number of experiments evaluating the quality of the analyses predicted by the model. We also describe how varying the way the model interprets rests in the input music affects the resulting analyses produced.

1. INTRODUCTION

Music analysis is primarily concerned with studying the structure of music compositions, both at the small- and large-scale levels. Hierarchical music analysis, best exemplified by *Schenkerian analysis*, illustrates the structure of a music composition by identifying hierarchical relationships among the notes of the music. These relationships collectively group the notes into a series of hierarchical levels that demonstrate the function of each note in the music in relation to other notes at various levels of the hierarchy.

One of the complicating factors of Schenkerian analysis is that there is no single established algorithm for performing the analysis. Instead, textbooks present guidelines and sample analyses from which students gradually learn the techniques, often through trial and error. Historically, there have been a number of research endeavors that attempted to replicate the Schenkerian analysis procedure: purely algorithmic efforts run into problems because of the

conflicting and ambiguous nature of the Schenkerian analysis rule set [4, 6] and up until recently, machine learning approaches often hit roadblocks due to the lack of a large standardized corpus of Schenkerian analysis upon which to train [5, 10, 11].

However, more recent efforts to create such a corpus of Schenkerian analysis have led to a data-driven system capable of learning to analyze music in a hierarchical fashion [7, 9]. This system, however, is only capable of hierarchically analyzing the monophonic main melody of the composition, with any other voices or harmonic parts contributing only auxiliary information to the algorithms. In this work, we study the practicality of extending this monophonic model of music analysis to support homophonic textures with a soprano part and a supporting bass line. We present evidence that there are homophonic patterns that can be harnessed by machine learning techniques, demonstrate the workings of a probabilistic model on homophonic input, and evaluate the system both for accuracy and for determining where mistakes are made.

Because Schenkerian analysis is one of the most comprehensive forms of music analysis available today [3], the uses of this work extend beyond the obvious application of studying computationally-produced analyses of music. Algorithms for calculating music similarity or identifying musical styles or genres could be enhanced with the probabilistic model described here, as could systems for music recommendation or new music discovery. At a more fundamental level, studying computational models of music analysis can lead us towards a better understanding of musical perception and structure [1].

2. MODELING MONOPHONY AND HOMOPHONY

Schenkerian analysis hypothesizes that music compositions are structured as a series of hierarchical levels defined by *prolongations*: situations where a note, chord, or melodic interval remains in control of a passage of music even though it may not be sounding constantly during that time. Consider the five-note descending phrase in Figure 1, occurring over G major harmony. In this melody, a music theorist would identify a prolongation over the first three notes D–C–B: the note C prolongs the passing motion from the D to the B. A similar prolongation occurs among the notes B–A–G. This places the notes D, B, and G — the notes being prolonged — at a higher structural



© Phillip B. Kirlin and David L. Thomas.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Phillip B. Kirlin and David L. Thomas. "Extending a Model of Monophonic Hierarchical Music Analysis to Homophony", 16th International Society for Music Information Retrieval Conference, 2015.

level than the C or the A.



Figure 1. An arpeggiation of a G-major chord with passing tones. The slurs are a Schenkerian notation used to indicate the locations of prolongations.

However, there is another level of prolongation at work in this melody. The interval of a fifth between the first note D and the last note G is prolonged by the motion to and from the B in the middle. This leads to a three-level hierarchy of intervals as shown in the binary tree in Figure 2(a). An equivalent structure, illustrated in Figure 2(b), is known as a *maximal outerplanar graph* or *MOP*: this structure is equivalent to a binary tree of melodic intervals but represents the same information more succinctly [14]. We claim that any hierarchical analysis can be represented as a MOP, and therefore, illustrated as a fully-triangulated polygon.

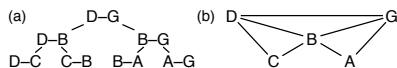


Figure 2. The prolongational hierarchy of a G-major chord with passing tones represented as (a) a tree of melodic intervals, and (b) a MOP.

By combining a corpus of musical excerpts and corresponding MOP analyses with a supervised machine learning algorithm, it is possible to learn a probabilistic model over MOP structures. This model admits a $O(n^3)$ algorithm for determining the most probable MOP for a new piece of music [9].

In the original MOP model of prolongation, a single triangle describes the elaboration of a parent melodic interval by two child intervals. This model, as first conceived, can only represent monophonic note sequences. As it would be desirable to enable hierarchical music analysis of all the voices within a composition, it is worth exploring extensions to represent multi-voice musical textures. One possibility is using a separate MOP to represent the structure of each voice in the music: this would allow for independent analyses of each voice. This representation, however, would increase the computational complexity of the algorithm for determining the most probable MOP analysis from $O(n^3)$ to $O(n^6)$ for a two-voice composition [13].

Instead, we investigate MOPs that store multiple pitches in a single vertex. In particular, we study MOPs that store up to two pitches per vertex, with the pitches derived from separate soprano and bass voices. We call these new MOPs *interval MOPs*, so named because the two pitches stored in a vertex form a harmonic interval between the soprano and bass parts. Where it is necessary to differentiate between the two varieties of MOPs, we will call the original type of MOP a *monophonic MOP*.

2.1 The Interval MOP Model

Consider the five-note descending melodic pattern from Figure 1, now augmented with a bass line, as in Figure 3(a). The equivalent interval MOP is shown alongside, in Figure 3(b). Clearly, the triangles within an interval MOP have the same prolongational interpretations as in monophonic MOPs. One will observe, however, that there can be potential conflicts in the prolongational structure between different voices. For instance, consider the melodic voices in Figure 4(a), where the prolongational slurs imply that the MOP-like structure in Figure 4(b) is necessary to represent the prolongations in the soprano and bass parts. Unfortunately, the definition of a MOP prohibits any edge-crossings of this variety: such a crossing breaks the strict hierarchy necessary to maintain the mathematical (and as we will show later, computational) properties inherent in a MOP.

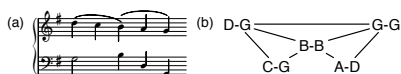


Figure 3. (a) A musical passage with independent soprano and bass parts, and (b) the corresponding interval MOP.

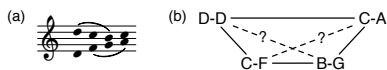


Figure 4. (a) A musical passage with conflicting soprano and bass prolongations and (b) the only way of representing both prolongations in a MOP-like structure, illustrating the conflicting edges that would arise.

Though note-against-note textures are easily represented in interval MOPs, some explanation is necessary for how to handle more complicated rhythms. When one voice has a change in pitch while another voice has a sustained pitch (i.e., oblique motion), the sustained note may be duplicated in the interval MOP. For example, consider the first and second beats in Figure 3(a): the bass note G is held for both of these beats and is duplicated in the first two vertices of the interval MOP in Figure 3(b).

Rests can be explicitly represented in an interval MOP. Suppose that the half note G in the bass part of of Figure 3(a) were a quarter note followed by a quarter rest. This would alter the interval MOP of Figure 3(b) to have the vertex C-G store the pair C-(rest) instead.

Every interval MOP contains two additional vertices, representing the *START* and *FINISH* of a composition. The *START* vertex is always (temporally) the first vertex in a MOP: it does not correspond to the first note in the music, but rather should be thought of as occurring before the start of the music. Similarly, the *FINISH* vertex is always the last vertex in a MOP and temporally occurs after the end of the music. These extra vertices are necessary to permit any pair of intervals in a MOP to represent the most

abstract level of the musical hierarchy. Consider a MOP not containing such extra vertices: because prolongations are oriented temporally, with the left and right vertices of a prolongation always higher in the structural hierarchy than the middle vertex, the most abstract edge of the MOP would have to be between the first harmonic interval of the music and the last. Because the first and last harmonic intervals are not always the most musically important pair of events, including START and FINISH allow for any pair of harmonic intervals in the music to represent the most abstract level in the structural hierarchy [14].

In the remainder of this paper, we study the feasibility of using interval MOPs to represent Schenkerian analyses for two-voice homophonic compositions. We do not consider polyphonic textures with completely independent voices due to the likelihood of encountering conflicting prolongational structures, such as in Figure 4(a), and the inability of interval MOPs to represent such structures, as discussed earlier. We show that there are patterns that arise in the encoding of music analyses in the interval MOP structure, we illustrate algorithms for harnessing these patterns and identifying the probabilistically most likely interval MOP analysis for new pieces of music, and we conclude with experiments showing how (1) accurately these algorithms can reproduce ground truth analyses and (2) what sorts of errors the algorithms make.

3. CONSTRUCTING INTERVAL MOPS FROM REAL-WORLD ANALYSES

Earlier work in computational Schenkerian analysis has verified that there are regularities in the prolongations that humans identify during the analysis procedure. Specifically, if we recall that each triangle in a MOP corresponds to a three-note prolongation, then it has been shown that various types of triangle occur more frequently than others [9]. However, in order to confirm this finding for interval MOPs, we first require an algorithm to convert a pair of monophonic MOPs — one representing the soprano line and one representing the bass line — into a single interval MOP. The strategy we use is to first align the notes of the monophonic MOPs to create an initial completely-untriangulated interval MOP consisting of corresponding pairs of notes between the soprano and bass MOPs. A pair of notes is created any time there is a temporal overlap between a soprano note and a bass note, so an individual note may appear multiple times in an interval MOP. Next, interior edges are added from the original soprano MOP in corresponding locations in the interval MOP; this has the same effect as copying every prolongation from the soprano MOP to the interval MOP. Lastly, all edges are added from the original bass MOP to the interval MOP that can be added without creating conflicts (overlapping edges). We prioritize the soprano prolongations because the soprano voice is more easily heard in the overall music and usually is more melodically significant.

We ran an experiment to verify the appropriateness of using interval MOPs as a representation of a multi-voice Schenkerian analysis. We used an updated version of the

SCHENKER41 corpus: a data set containing 41 excerpts of common practice period music and corresponding Schenkerian analyses. All of the music in the corpus is for a solo keyboard instrument or for voice with keyboard accompaniment, is in a major key, and does not modulate. All of the excerpts are between two and sixteen measures in length, but most are either four or eight measures long. 39 of the Schenkerian analyses in the corpus are taken from textbooks and two analyses were sourced from a local expert music analyst [7]. We translated all the musical excerpts from monophonic MOPs to interval MOPs using the algorithm described above. Because we are interested in confirming that there are patterns in prolongational data as represented by interval MOPs, we examined how often every type of triangle appeared in the converted interval MOPs.

Specifically, we calculated the frequencies of all triangle types in the corpus in order to test the statistical significance given the null hypothesis that the corpus analyses represented as interval MOPs resemble randomly-constructed MOPs in their triangle frequencies. Determining the expected frequency of a triangle in a MOP under this null hypothesis is straightforward precisely because of the mathematical underpinnings of the MOP formulation.

Assume we have a polygon with n vertices, numbered clockwise from 0 to $n-1$, and we are interested in the number of times that the triangle between vertices x , y , and z ($x < y < z$) appears across all complete triangulations of this polygon. We observe that any triangle drawn inside a polygon necessarily divides the interior of the polygon into four regions: the triangle itself, plus the three regions outside the triangle but inside the polygon, as in Figure 5 (though it is possible for some of these regions to be degenerate line segments). Any complete triangulation of the polygon that contains $\triangle xyz$ must necessarily completely triangulate the three regions outside of the triangle, and we simply multiply the number of ways of triangulating each of those three regions to obtain the total number of complete triangulations that contain $\triangle xyz$.

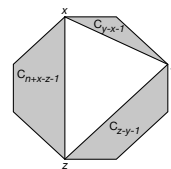


Figure 5. The number of times $\triangle xyz$ appears in all possible triangulations of the octagon can be calculated from the sizes of the shaded regions.

The number of ways of triangulating each of the three regions is directly related to the size of each region, which we can calculate from the values of the vertices x , y , and z . The sizes (number of vertices in the polygons) of these regions are $y - x + 1$, $z - y + 1$, and $n + x - z + 1$, respectively. Precisely because edges in MOPs cannot cross, the number of triangles that will appear in each of these

regions is solely a function of the size of each region: the number of ways to triangulate each region is the Catalan number for the size of each region minus two, and therefore the complete calculation for the expected frequency is $C_{y-x-1} \cdot C_{z-y-1} \cdot C_{n+x-z-1}$, where $C_i = \frac{1}{i+1} \binom{2i}{i}$.

We ran binomial tests for each type of triangle by comparing the expected frequencies of the triangles with the observed frequencies in the corpus. We found that there were 48 different types of triangles that were possible in the corpus of interval MOPs, where a triangle type was defined by categorizing the three harmonic intervals between the endpoints as either *consonant*, *dissonant*, *single* (for single notes), or *not applicable* (for MOP vertices containing a START or FINISH vertex). We checked for triangles that were statistically significant at the 5% level. By using the Šidák correction, we found that only triangles that had a p -value of less than 0.001 would be considered significant; there were six triangles that matched this criteria. These triangles are described in Figure 6.

4. A PROBABILISTIC INTERPRETATION OF INTERVAL MOPS

Now that we have verified that there are statistically significant prolongational patterns in the corpus of interval MOPs, we may continue towards our goal of developing an algorithm to harness the patterns in such a way as to be able to analyze new compositions. We proceed in a manner similar to that which was used in the original probabilistic model of monophonic MOPs [9].

Given two monophonic sequences of notes, a soprano line $S = s_1, s_2, \dots, s_n$, and a bass line $B = b_1, b_2, \dots, b_m$, our goal is to calculate the most probable analysis A for these notes, which means maximizing $P(A | S, B)$. An interval MOP is defined by the set of triangles T_1, T_2, \dots, T_k within, and thus we define

$$P(A | S, B) = P(T_1, T_2, \dots, T_k).$$

This full joint probability distribution cannot be efficiently estimated using the amount of training data available to us, so we decompose it into a product of probabilities of individual triangles:

$$P(T_1, T_2, \dots, T_k) \approx P(T_1) \cdot P(T_2) \cdots P(T_k).$$

In other words, we assume that each triangle in an interval MOP is independent of the other triangles. An earlier experiment [8] verifies that this does not appreciably alter the probabilistic rankings of the MOP analyses.

We define the individual probability of a triangle within an interval MOP analysis in terms of random variables representing the three endpoints of the triangle:

$$P(T_i) = P(C_i | L_i, R_i).$$

The three random variables in this distribution each represent either a harmonic interval or a single soprano or bass note with a rest in the other voice. The endpoints are unambiguously named because MOPs are oriented by the temporal dimension: later notes always appear to the right of earlier notes.

Our goal is to use the SCHENKER41 data set to estimate $P(C | L, R)$, but this is impractical due to the high-dimensional nature of the random variables involved: we would like to use melodic, harmonic, and rhythmic features of the triangle endpoints, and a data set of 41 analyses does not give us enough data to do this by directly counting triangle frequencies and normalizing them into a probability distribution. Instead, we use random forests [2], a type of ensemble classifier, to estimate this probability. Specifically, we create a large collection of decision trees, with each tree designed to predict a certain feature of the middle point C , trained on a subset of the features of the left and right endpoints L and R . The predictions of all the trees for a given feature of C are then aggregated and normalized into a probability distribution [12].

4.1 Features

We use a set of twenty-seven features to represent a triangle. Specifically, we use eighteen features solely involving the left and right endpoints (L and R) to predict nine features for the center point (C). These features are:

- The category of the interval involving the soprano and the bass note, listed either as *Cons* (Consonant) or *Dis* (Dissonant) (three features, one each for L , C , and R).
- For a given note in an interval, the scale degree (1-7) of the note (six features).
- The harmony present in the music at the time of the interval as a Roman numeral (six features). These harmonic labels, provided by experts, are included in the SCHENKER41 corpus.
- The broader category of harmony present in the music at the time of the interval, such as *tonic* or *dominant* (six features).
- For a given note in an interval, whether the note was a chord tone in the harmony present at the time (six features).

In some situations, certain features are not applicable.

In the case that L or R is a START or FINISH vertex, the features are marked with invalid values to denote their ineligibility. Furthermore, in situations where L , C , or R is not an interval, but instead a single note, only half of the attributes per category listed above are applicable.

5. EVALUATION

As mentioned earlier, one reason for preferring interval MOPs to a more complicated representation for multi-voice prolongational hierarchies is the mathematical elegance of the structure, which makes it an efficient choice from which to infer probabilistic patterns. No less important is fact that computing the optimal triangulation of a polygon can be done in $O(n^3)$ time by using a standard dynamic programming algorithm. This is the basis of the existing PARSEMOP algorithms designed for monophonic MOPs; we adapt the algorithms to work with interval MOPs.

There are three variations of PARSEMOP; each variation is given different amounts of *a priori* information regarding the most abstract level of the hierarchical analysis

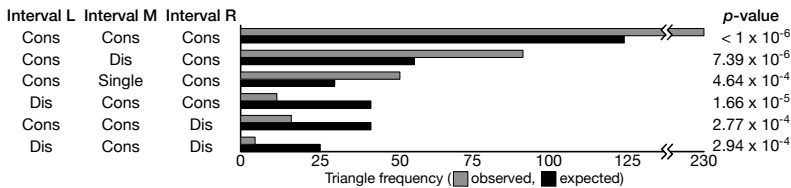


Figure 6. Types of triangles statistically significant at the 5% level.

being produced. Heinrich Schenker theorized that all tonal music compositions were derived from one of a small set of simple structures involving a short melodic progression harmonized in a specific way. Thus, the Schenkerian analysis process of finding prolongations theoretically always reveals one of these structures, known as the *fundamental structure* or *Ursatz*, at the background level.

All variants of the PARSEMOP algorithm accept the musical score as input, and are told which notes of the score constitute the soprano and bass lines. PARSEMOP-A has no conception of the *Ursatz* built into the algorithm, and therefore will not necessarily find one of the fundamental structures in the music when it runs. PARSEMOP-B, on the other hand, in addition to the musical score, is also informed as to which specific notes in the score should be placed into the background fundamental structure. This version of PARSEMOP, therefore, will always find the correct background structure. PARSEMOP-C is a compromise between the structurally-unaware PARSEMOP-A, and the overly-aware PARSEMOP-B: this version is informed as to which musical *itches* constitute the fundamental structure and in what order they should appear in the output, but the algorithm is not told the exact locations of the corresponding notes in the score.

We used leave-one-out cross-validation in conjunction with the SCHENKER41 corpus to evaluate how well the three PARSEMOP algorithms could reproduce the ground-truth analyses in the corpus. Specifically, for each of the 41 excerpts in the corpus, we trained our probabilistic model on the interval MOPs derived from the other 40 excerpts, and then used each PARSEMOP algorithm to derive the most probable analysis for the original piece omitted. We compared the algorithmically-produced MOPs to the ground-truth MOPs using an metric called *edge accuracy*, which is the proportion of internal edges in an interval MOP that correspond to an edge in the ground-truth interval MOP. We use this metric rather than proportion of triangles that match between two analyses because there are cases where two analyses can have edges in common, indicating some similarity, yet have no triangles in common.

Although occasionally music analysts may disagree on what the “correct” Schenkerian analysis should look like for a piece of music, the limited amount of data allowed us only one ground-truth analysis per musical excerpt.

Figure 7 shows the aggregate edge accuracy levels for the three PARSEMOP algorithms. For the sake of compar-

ison, we included the average edge accuracy as would be obtained by a baseline algorithm that analyzes music randomly: this hypothetical algorithm creates triangulations uniformly at random from the space of all possible complete triangulations.

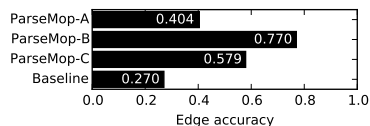


Figure 7. Edge accuracies for the three PARSEMOP algorithms and the baseline randomized algorithm.

Interestingly, the accuracy levels obtained by analyzing all possible analyses for a given piece of music do not follow a uniform or normal distribution. In fact, the distribution of edge accuracies as would be obtained by selecting a complete triangulation uniformly at random is quite skewed, as can be seen in Figure 8. This means that even though the PARSEMOP algorithms never break 80% accuracy, when compared against the baseline algorithm, they are doing quite well. In fact, we can use the distribution of edge accuracies from the baseline algorithm to judge each PARSEMOP algorithm’s accuracy against the null hypothesis that the PARSEMOP algorithm does no better than random. This results in *p*-values of 0.1022, < 0.0001 , and 0.0061 for the -A, -B, and -C varieties of PARSEMOP, respectively.

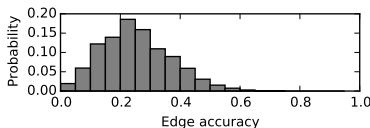


Figure 8. Distribution of edge accuracies under the baseline random algorithm.

We also analyzed where in the algorithmically-produced MOP analyses PARSEMOP was making mistakes. Specifically, for each non-perimeter edge in a PARSEMOP analysis that did not correspond to an edge in the ground MOP, we computed the *edge depth*: a number between 0 and 1 indicating how far down in the structural hierarchy the edge lies, with 0 being the most abstract level of the hierarchy

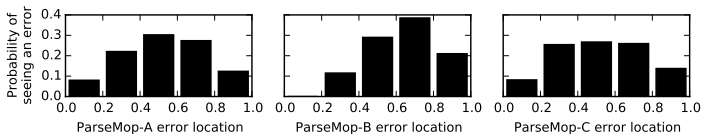


Figure 9. Probability distributions of the locations of errors in the PARSEMOP analyses.

and 1 being the surface level of the music. We produced probability distributions illustrating the hierarchical locations where PARSEMOP is most likely to make an error; these are shown in Figure 9. These distributions are unsurprising: PARSEMOP-A and -C make fewer errors at the extreme levels of the hierarchy due to the surface-level music constraining the low-level decisions at one end and fewer high-level decisions to be made at the other. Furthermore, PARSEMOP-B makes fewer mistakes at the most abstract level because the *Ursatz* has been supplied ahead of time.

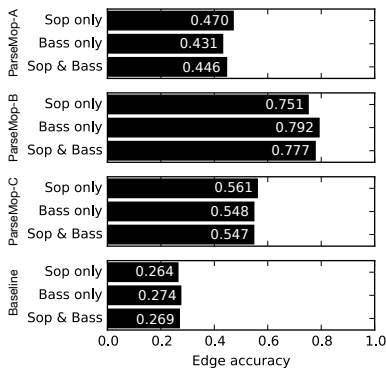


Figure 10. Edge accuracies after three varieties of rest adjustment for the three PARSEMOP algorithms and the baseline randomized algorithm.

6. ACCOUNTING FOR RESTS

Though the results from the previous section indicate that the PARSEMOP algorithms using interval MOPs are doing relatively well when compared against the baseline algorithm, there is still plenty of room for improvement. One area we hypothesized that could be adversely affecting accuracy is the presence of rests in the soprano and bass parts. Recall that each vertex in an interval MOP holds both a soprano and bass note, but only for note pairs sounding at the same time. Notes may be paired with rests if a rest is “sounding” at the same time as a note in the other voice. This may be the wrong musical interpretation, however, in situations where rests are stylistic indications for performance (e.g., a substitute for staccato markings), rather than indications that a melodic line contains a true pause. Thus, we present a modification to the interval MOP construction algorithm that within a voice, extends each note through

any intervening rests up to the start of the next note within a voice. In essence, all rests are eliminated from the soprano and bass parts, and notes durations are increased to fill the gaps. There are three different versions of the rest adjustment algorithm that control which voices are adjusted: just the soprano, just the bass, or both voices adjusted.

The rest adjustment algorithm, when applied to all of the soprano lines in the corpus, modifies the durations of 102 notes out of a total of 931. When applied to the bass line, the algorithm elongates 316 notes out of a total of 908.

We re-ran the earlier cross-validation experiment with each of the three versions of the rest adjustment algorithm; the updated edge accuracies are shown in Figure 10. Interestingly, the only situation in which the rest adjustment algorithm had any large affect on the edge accuracy was for PARSEMOP-A, where it increased the edge accuracy from roughly 40% to between 44% and 47%. Effects on PARSEMOP-B and PARSEMOP-C were much smaller, and in some cases caused a slight decrease in accuracy. The effects on the p -values under the null hypothesis that the PARSEMOP analyses resemble analyses chosen at random were also small; these new p -values are shown in Table 1.

PARSEMOP variant:	A	B	C
Sop only	0.0420	< 0.0001	0.0090
Bass only	0.0853	< 0.0001	0.0133
Sop & Bass	0.0601	< 0.0001	0.0110

Table 1. p -values calculated under the null hypothesis that PARSEMOP analyses (with the rest adjustment algorithm) resemble analyses done randomly.

7. DISCUSSION

Overall, the results from this study are encouraging. The edge accuracies and their improvement over the random baseline algorithm imply that interval MOPs can successfully model a homophonic prolongational hierarchy. Interval MOPs maintain all of the mathematical and computational advantages of monophonic MOPs, including a straightforward learning algorithm and a computationally-efficient method for finding the most probable analysis for a new piece of music.

However, it is clear that interval MOPs cannot represent all of the prolongational situations that could arise in polyphonic textures, namely conflicting prolongations between voices. We plan on studying the feasibility of using independent MOPs for the soprano and bass; this will alleviate the representational issue, but may require an approximation algorithm for finding the most probable MOPs for new compositions in order to remain computationally tractable.

8. REFERENCES

- [1] Samer A. Abdallah and Nicolas E. Gold. Comparing models of symbolic music using probabilistic grammars and probabilistic programming. In *Proceedings of the Joint 11th Sound and Music Computing Conference and 40th International Computer Music Conference*, pages 1524–1531, 2014.
- [2] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.
- [3] Matthew Brown. *Explaining Tonality*. University of Rochester Press, 2005.
- [4] R. E. Frankel, S. J. Rosenschein, and S. W. Smoliar. Schenker’s theory of tonal music—its explication through computational processes. *International Journal of Man-Machine Studies*, 10(2):121–138, 1978.
- [5] Édouard Gilbert and Darrell Conklin. A probabilistic context-free grammar for melodic reduction. In *Proceedings of the International Workshop on Artificial Intelligence and Music, 20th International Joint Conference on Artificial Intelligence*, pages 83–94, Hyderabad, India, 2007.
- [6] Michael Kassler. Proving musical theorems I: The middleground of Heinrich Schenker’s theory of tonality. Technical Report 103, Basser Department of Computer Science, School of Physics, The University of Sydney, Sydney, Australia, August 1975.
- [7] Phillip B. Kirlin. A data set for computational studies of Schenkerian analysis. In *Proceedings of the 15th International Society for Music Information Retrieval Conference*, pages 213–218, 2014.
- [8] Phillip B. Kirlin and David D. Jensen. Probabilistic modeling of hierarchical music analysis. In *Proceedings of the 12th International Society for Music Information Retrieval Conference*, pages 393–398, 2011.
- [9] Phillip B. Kirlin and David D. Jensen. Using supervised learning to uncover deep musical structure. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, pages 1770–1776, 2015.
- [10] Alan Marsden. Schenkerian analysis by computer: A proof of concept. *Journal of New Music Research*, 39(3):269–289, 2010.
- [11] Panayotis Mavromatis and Matthew Brown. Parsing context-free grammars for music: A computational model of Schenkerian analysis. In *Proceedings of the 8th International Conference on Music Perception & Cognition*, pages 414–415, 2004.
- [12] Foster Provost and Pedro Domingos. Tree induction for probability-based ranking. *Machine Learning*, 52(3):199–215, September 2003.
- [13] Hiroyuki Seki, Takashi Matsumura, Mamoru Fujii, and Tadao Kasami. On multiple context-free grammars. *Theoretical Computer Science*, 88(2):191–229, 1991.
- [14] Jason Yust. *Formal Models of Prolongation*. PhD thesis, University of Washington, 2006.

THE MIR PERSPECTIVE ON THE EVOLUTION OF DYNAMICS IN MAINSTREAM MUSIC

Emmanuel Deruty

Sony Computer Science Laboratory / Akoustic Arts
Paris, France
emmanuel.deruty@gmail.com

François Pachet

Sony Computer Science Laboratory
Paris, France

ABSTRACT

Understanding the evolution of mainstream music is of high interest for the music production industry. In this context, we argue that a MIR perspective may be used to highlight, in particular, relations between dynamics and various properties of mainstream music. We illustrate this claim with two results obtained from a diachronic analysis performed on 7200 tracks released between 1967 and 2014. This analysis suggests that 1) the so-called “loudness war” has peaked in 2007, and 2) its influence has been important enough to override the impact of genre on dynamics. In other words, dynamics in mainstream music are primarily related to a track’s year of release, rather than to its genre.

1. INTRODUCTION

Mainstream popular music is in constant evolution. There may be more differences than common points between progressive rock albums from the 1970’s such as Pink Floyd’s best-selling “Dark Side of the Moon” and contemporary rap albums such as Nicki Minaj’s platinum-certified “Roman Reloaded”. Studies tracking down the yearly evolution of signal descriptors are useful to characterize this diversity.

In 1982, Moller [1] established that recent recordings feature a larger dynamic excursion than older ones. More recently, Tardieu [2] studied the evolution of stereo, dynamic and spectral features on pop/rock songs, and showed that decade classification accuracies using spectral and dynamic features are equal. Pestana [3] focused on spectral features and found that while spectra are dependent on genre, they also follow the yearly evolution of production standards. Serrà [4] performed a systematic analysis of more than 400,000 tracks and concludes that popular music “show[s] no considerable changes in more than fifty years” other than becoming louder, a result challenged by Mauch [5]. Deruty [6] focused on the changes in loudness and dynamics over the same period, and provided a characterization of the phenomenon referred to as the “loudness war”. The loudness war, or loudness race, is a trend in popular music production that affects mainstream music dynamics [7]. It has been de-

scribed as a contest between bands and record companies, in which music is engineered to be louder than the competition’s [8, pp. 237–292]. Starting at the end of the 80’s [4], [6], [9], its effects have been spectacular enough to reach the general media [10]–[11]. A distinction is made between dynamics occurring on different time-scales. The large-scale variations are known as macrodynamics, whereas the faster variations are referred to as microdynamics [12]–[13]. The loudness war favors high loudness tracks with reduced microdynamics [4], [6], [9], although some authors claim it has also reduced macrodynamics [14]. Efforts have been made to reverse the trend, through measurement protocols [15]–[16], integrated loudness-leveling engines such as iTunes’ Sound Check [17], or public communications [18]–[19].

In this paper, we perform a diachronic analysis on 7200 mainstream tracks released between 1967 and 2014, and present two results. First, we show that the evolution towards louder and less dynamic content peaked in 2007, and then started to decrease. If this trend continues, pre-loudness war values for most descriptors of music dynamics may be observed sometimes between 2017 and 2026. Second, we demonstrate that the loudness war’s impact supersedes the influence of music genre on dynamics. In mainstream music, a piece’s dynamics are more typical of a given year than they are of a given genre.

2. METHOD

2.1 Music corpus

The music corpus we rely on is a revision and extension of the corpus used in [6]. It includes 7200 tracks released between 1967 and 2014, 150 tracks per year. Track selection is based on Besteveralbums.com, a review aggregator. For each year, we choose the albums with the best ratings. If a given artist is the author of more than three well-rated albums, we select the artist’s complete discography. While this method does not lead to a random sampling, it ensures that the corpus is based on music that is popular. We choose to start the corpus at the end of the sixties because these years can be considered as the advent of the contemporary pop/rock era, characterized by the creative use of the recording studio [8, p. 157] along with mass media availability [20].

2.2 Signal descriptors

We use the signal descriptors defined in [6]. The track’s physical level is measured using the RMS power of the signal after normalization. Track loudness evaluation is



© Emmanuel Deruty, François Pachet.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Emmanuel Deruty, François Pachet. “The MIR perspective on the evolution of dynamics in mainstream music”, 16th International Society for Music Information Retrieval Conference, 2015.

performed using the EBU3341 integrated loudness [21], which has been shown to be as robust as more complex measures such as detailed perceptual models [22]. Microdynamics are measured using a variation of the crest factor, as defined in [6]. For macro-dynamics, we rely on the EBU3342 Loudness Range [23], which is, to our knowledge, the only normative descriptor to quantify dynamics in a musical sense (*piano*, *forte*...) [9]. We evaluate the overall amount of dynamic processing using the Peak to RMS Regression Coefficient (PRRC). PRRC values below 1 indicate usage of dynamic compression, values above 1 usage of dynamic expansion [6]. Finally, we estimate the amount of limiting applied to the tracks using the High Level Sample Density (HLSD) [6]. HLSD can be linked to the practice of limiting [12], which is suspected to have a decisive impact on mainstream music production during the last 30 years [8, pp. 237-292], [9], [14], [24]–[27]. Using relations between limiting and HLSD as shown in [6], we indeed find that a significant amount of limiting (> 3dB) seems to have been applied on 33% of all tracks from our corpus, and on 65% of tracks released after 1994.

For each descriptor, we provide a projection based on the current trend by fitting the descriptor’s smoothed median values using a second-degree polynomial, starting from the year for which the loudness war is observed to peak. As illustrated in Figure 1 (black dot at the right of the graphs), estimation of the return to pre-loudness war values is obtained using the crossing of the projected values with the median of the pre-1990 descriptor values.

2.3 Genre labels

Following [28]–[30], we draw the music genre labels from AllMusic, a website that provides “unoptimized expert annotated ground truth dataset for music genre classification” [30] in the form of a database of commercial music annotated in terms of “genres”, “meta-styles” and “styles”. Whereas AllMusic provides only 21 “genres”, album information also comes with 905 “styles” and “meta-styles” that can be interpreted as sub-genres to refine the major genre labels. In this paper, while relying on the “styles” provided by Allmusic, we designate them as “genres”, “a conventional category that identifies pieces of music as belonging to a shared tradition or set of conventions” [31]. Under this terminology, the 7500 tracks from the corpus correspond to 272 distinct mainstream music genres, each track being associated with a mean of 4 genres, the minimum being 1 and the maximum 11. Conversely, each genre is represented with a mean of 110 tracks, the minimum being 3 and the maximum 2482. Issues linked to the pertinence of the results regarding this diversity of representation are discussed in Section 4.3.

3. DIACHRONIC STUDY OF DYNAMICS

Figure 1 illustrates the descriptors’ behavior over time. The boxes’ upper and lower limits indicate the 25th and 75th percentiles of the distribution. The darker box indicates the peak of the loudness war for the descriptor, i.e. the year for which the median value is maximal. The

small horizontal lines inside the boxes indicate the median. The outer whiskers stand for the 5th and 95th percentiles. The solid, thick black curve is the smoothed median, on which the projection is based. The projection itself is represented by a dashed gray line. The thin horizontal line indicates the median pre-1990 descriptor values.

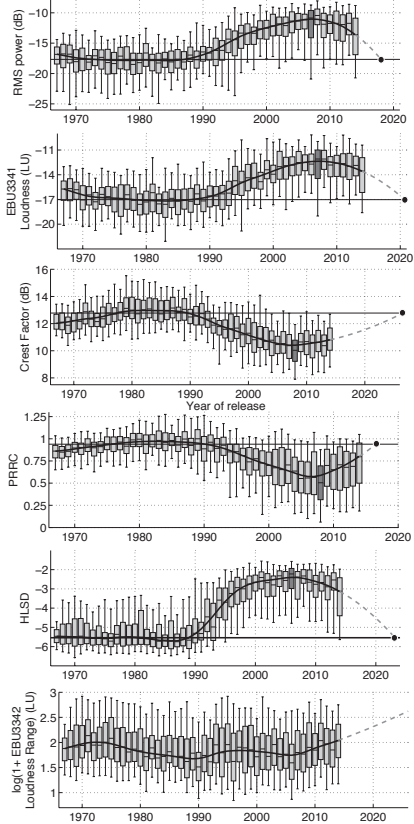


Figure 1. Descriptor evolution over the years. From top to bottom RMS power, EBU3341 integrated loudness, crest factor, PRRC, HLSD and EBU3342 Loudness Range.

The loudness war may be characterized by a change towards previously unobserved descriptor values that starts around 1990 and indicates the use of more dynamic compression [6]. Table 1 summarizes the loudness war timeline depending on the descriptor. It took ca. 15 years

for the loudness war to peak. The return to pre-loudness war values could take between 10 and 20 years. Figure 1 shows that macrodynamics are not affected by the loudness war. No significant change of values starting around 1990 and pointing toward more dynamic compression can be observed. The loudness war has increased music level and micro-dynamics, but has not decreased macro-dynamics.

Descriptor	Corresponding phenomenon	Peak	Estimated return to pre-loudness war values
RMS power	Physical level	2007	2018
EBU3341	Loudness [21]	2007	2020
Crest factor	Microdynamics [6], [12]	2008	2026
PRRC	Overall amount of dynamic processing [6]	2008	2017
HLSL	Amount of limiting [6]	2006	2023

Table 1. Loudness war timeline summary.

Since 2006, macrodynamics have increased consistently, and are higher in 2014 than they have ever been during the time-span covered by the corpus. This increase can be put in relation with a demand for more dynamics combined with the confusion that's often made between micro- and macrodynamics [6], [10]–[11], [14], [32]. Musicians and producers may be trying to counter the effects of the loudness war by raising macrodynamics, whereas raising microdynamics would be more productive in that respect. However, examination of Figure 1 shows that macrodynamics follow relatively shorter trends than other descriptors, and a reversal of the present tendency towards less macrodynamics could be witnessed as soon as 2015.

4. DYNAMICS AND MAINSTREAM GENRES

4.1 Dependency of dynamics on genres and trends

In this section, we show that dynamics of mainstream music are more typical of a given year than they are of a given genre. Figure 2 illustrates the distribution of RMS power values depending of the music genre of the track. On first approach, it suggests that music genre and RMS power are related. However, as illustrated in Figure 1, RMS power is also related to the year of the album release. Figure 3 provides more details, by illustrating RMS power evolution for the four most represented genres in the corpus (Alternative Pop/Rock, Alternative/Indie Rock, Album Rock and Contemporary Pop/Rock). It indicates that genres follow the year's trend in terms of RMS power. This phenomenon, previously mentioned in [32], suggests that RMS values may be primarily related to the year of the track release, rather than to its genre. We use two methods to confirm the tendency: a standard ANOVA and a variance evaluation.

The second method possesses the advantage of providing results formulated using the original descriptor's unit, and therefore being easier to interpret than the ANOVA's results. It involves the evaluation of the RMS distribution's variance for each genre and for each year, followed by the computation of the weighted arithmetic means of the variances, taking into account genre and year representativeness. The process is illustrated in Figure 4. The weighted mean variance for each year is 9.03dB, whereas the weighted mean variance for each style is 14.19dB. This shows that RMS values primarily originate from the track's year of release. In other words, particular physical levels are more typical of a given year than they are of a given genre. As shown in Table 2, this result is confirmed by the ANOVA's F -statistic. We repeat the experiment using the other descriptors described in Section 2.2. Results are similar. With the exception of the EBU3342 LRA, descriptors are clearly more related to the year's trend than to the piece's genre.

Descriptor	Mean variance for each year	Mean variance for each genre	ANOVA's F -statistic (years as classes)	ANOVA's F -statistic (genres as classes)
RMS power	9.03dB	14.2dB	107.7	6.4
EBU3341	4.57LU	7.41.LU	104.9	6.5
Crest factor	1.35dB	2.25dB	110.4	5.4
PRRC	0.04	0.06	77.5	4.9
HLSL	0.79	2.08	274.7	8.6
EBU3342	14.5LU	14.3LU	7.3	4.7

Table 2. Comparison of the weighted mean arithmetic means of the descriptor variances for each year and each genre, as well as comparison of the ANOVA's F -statistics, show that dynamics in mainstream music are primarily linked to the piece's year of release, rather than to its genre.

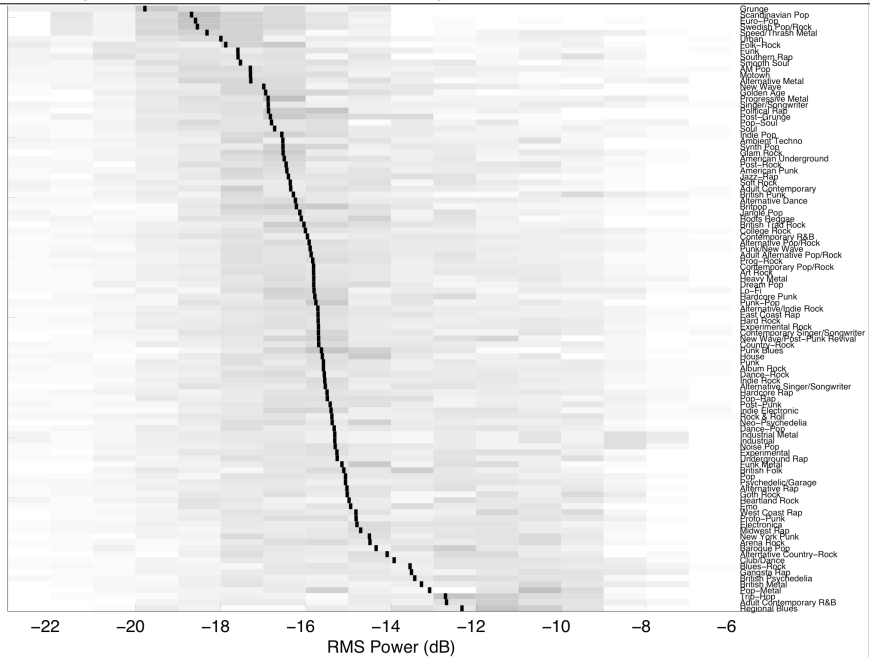


Figure 2. Distribution of RMS power values depending on the tracks’ genres. Darker shades of gray indicate higher levels of distribution. The black rectangles indicate the median. This Figure is restricted to styles corresponding to more than 50 tracks.

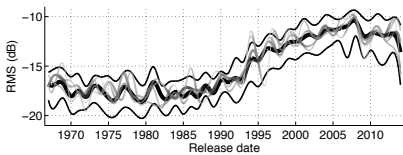


Figure 3. In gray, RMS power values corresponding to the music genres most represented in the corpus. Lighter gray sections indicate years with fewer tracks. The three black lines represent the 25th, 50th and 75th percentiles.

4.2 The particular cases of HLSD and LRA

As shown in Table 2, a particularly high dependence to trends is clear in the case of the HLSD, with an *F*-statistic being higher than in the case of the other descriptors. As seen in Section 2.2, it implies that the amount of limiting applied by audio engineers during mastering can be considered as independent from genre. Therefore, mainstream genres cannot be said to sound more or less “hot”. This is an important information in the context of mainstream music mastering: it can help engineers choose and argue the output level with their client, which is often a critical debate [33]. On the other hand, dependency to trends is much lower in the case of the EBU3342 LRA. As a result, macrodynamics can be considered as relatively independent from both genre and year of release.

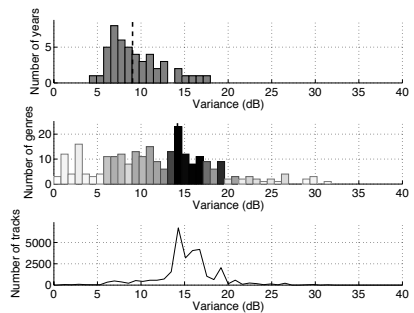


Figure 4. Distribution of RMS power variances. Top, by year. Middle, by genre. The dashed vertical line represents the weighted mean of the distribution. Bar hues indicates style representativeness. Bottom, style representativeness displayed quantitatively.

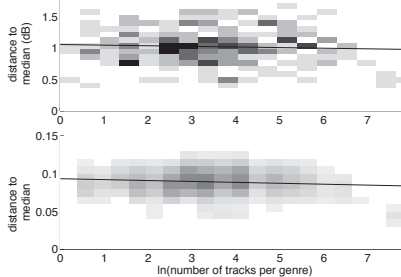


Figure 5. Top, distance between each genre and the all-genre median value for the RMS descriptor, against genre representation. Bottom, result of the same process using random values between 0 and 1. The horizontal line represents the linear regression.

4.3 Discussion

It may seem counter-intuitive to conclude that dynamics are more dependent on trends than they are on genres. Indeed, genres such as Euro-Pop exhibit high micro-dynamics and low overall loudness, whereas other genres such as Trip-Hop are associated with low micro-dynamics and high overall loudness. However, the Euro-Pop genre is most represented in the 1970s and 1980s [34], at a period when music was produced to feature high microdynamics and low overall loudness [6]. Trip-Hop is mainly a mid-1990s trend [35], a moment when low microdynamics and high overall loudness were common in music production [6]. Conversely, all genres that span several decades follow the trend of the year of production.

As mentioned in Section 2.3, not all genres are equally represented. This may bring the suspicion that dynamics are only dependent on the trends followed by the most represented genres, such as the subgenres of rock represented in Figure 3, but independent from the trends followed by most other genres, in which case our conclusion would not stand. To discard this suspicion, we evaluate the distance between each genre and the all-genre median value for the descriptors over the years. This distance is then matched against the genre's number of occurrences. Figure 5, top, illustrates the case of the RMS descriptor. A few well-represented genres are indeed closer to the median than most other genres. However, Figure 5, bottom, illustrates the same process using 1000 sets of 7500 random values in place of the 7500 RMS values. Both graphs are similar, and the few well-represented genres are closer to the median in both cases. Therefore, a particular dependency to a few genres is not a property of the present corpus. This discards the suspicion according to which the dependency to trends we found is only valid as far as a few genres are concerned.

5. CONCLUSION

Mainstream music dynamics are thought to be conditioned by genre, in terms of overall track loudness [36], microdynamics [9], [37], macrodynamics [15], [38] or amount of dynamic processing applied to music pieces during the production stage [7], [39, p. 121]. However, using a MIR perspective, we have shown that dynamics and overall loudness depend more on the track's year of release than on its genre. We have also found, as suspected by [40], that the loudness war has influenced all mainstream genres indiscriminately. A notable exception lies in macrodynamics as measured by the EBU3342 Loudness Range, which are more independent from both genre and year of release. In other words, dynamic range in the musical sense (*pianissimo* to *fortissimo*) is only marginally dependent on either mainstream genre or trend.

According to mastering engineer Bob Katz, the loudness wars were over in 2013 [41]. We have shown that the loudness war has peaked in 2007, and that a return to pre-loudness war dynamics may be reached in about ten years. As an exception, macrodynamics, which have not been significantly influenced by the loudness war, appear to increase since the loudness war's peak, and are currently reaching very high values.

This is useful knowledge in several situations. Many artists and producers ask sound engineers to increase loudness during mastering [33], arguing that the music genre to which their tracks belong is well suited to a "hot", loud and compressed sound. The present study provides objective data to challenge this claim. Loudness war activists argue for more important dynamics [32], [41]. We have shown that this concerns only microdynamics. Automatic mixing and mastering rely on constraints to be applied on initial audio content [42]–[44]. The present study has demonstrated that constraints relative to dynamics in mainstream music may be derived from trends rather than genres.

More generally, we suggest that the present method could be used for other audio descriptors, in order to establish their dependency to either diachronic trends, genre, or to any other musical dimension.

6. ACKNOWLEDGEMENTS

This research is supported by the project *Lrn2Cre8* which is funded by the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 610859.

7. REFERENCES

- [1] L.G. Moller, "How Much Has Amplitude Distribution of Music Changed?," in *AES 71st Conv.*, Montreux, Switzerland, 1982.
- [2] D. Tardieu *et al.*, "Production Effect: Audio Features For Recording Technique Description and Decade Prediction," in *14th Int. Conf. on Digital Audio Effects (DAFX 11)*, Paris, France, 2011.

- [3] P.D. Pestana *et al.*, "Spectral Characteristics of Popular Commercial Recordings 1950-2010," in *AES 135th Conv.*, New-York, NY, 2013.
- [4] J. Serrà *et al.*, "Measuring the evolution of contemporary western popular music," *Scientific Reports*, Jul. 2012.
- [5] M. Mauch *et al.* (2015). *The evolution of popular music: USA 1960-2010* [Online]. Available: <http://rsos.royalsocietypublishing.org/content/2/5/150081>. Royal Society Open Science, DOI: 10.1098/rsos.150081.
- [6] E. Deruty and D. Tardieu, "About Dynamic Processing in Mainstream Music," *J. Audio Eng. Soc.* 62(1/2), pp. 42-55, Jan. 2014.
- [7] E. Vickers, "The loudness war: Background, speculation and recommendations," in *AES 129th Conv.*, San Francisco, CA, 2010.
- [8] G. Milner, *Perfecting sound forever: an aural history of recorded music*. London, UK: Faber & Faber, 2009.
- [9] E. Deruty, "'Dynamic range' and the loudness war," *Sound on Sound*, Sep. 2011.
- [10] E. Smith, "Even heavy-metal fans complain that today's music is too loud!!!" *The Wall Street Journal*, Sep. 2008.
- [11] K. Matersonn, "Loudness war stirs quiet revolution by audio engineers," *Chicago Tribune*, Jan. 2008.
- [12] E. Skovenborg, "Measures of Microdynamics," in *AES 137th Conv.*, Los Angeles, CA, 2014.
- [13] B. Katz and R. Katz, *Mastering audio: the art and the science*. Focal Press, 2007
- [14] S. Sreedhar. "The future of music," *IEEE Spectrum*, Aug. 2007.
- [15] E. Skovenborg, "Loudness Range (LRA), Design and Evaluation," in *AES 132th Conv.*, Budapest, Hungary, 2012.
- [16] European Broadcasting Union. (2015). *Loudness* [Online]. Available: <https://tech.ebu.ch/loudness>.
- [17] T. Lund. (2013). *Audio for Mobile TV, iPad and iPod* [Online]. Available: <http://www.tcelectronic.com/media/2040040/mobile-test-paper-2013>.
- [18] J. V. Serinus. (2012). *Winning the Loudness Wars* [Online]. Available: <http://www.stereophile.com/content/winning-loudness-wars>.
- [19] R. Archer. (2014). *Have the Loudness Wars Peaked?* [Online]. Available: http://www.cepro.com/article/excuse_the_pun_have_the_loudness_wars_peaked/.
- [20] *The Cambridge companion to recorded music*. Cambridge, UK: Cambridge University Press, 1999.
- [21] European Broadcasting Union. (2011). *EBU - TECH 3341* [Online]. Available: <https://tech.ebu.ch/docs/tech/tech3341.pdf>.
- [22] E. Skovenborg and S. Nielsen, "Evaluation of different loudness models with music and speech material," in *AES 117th Conv.*, 2004, San Francisco, CA, 2004.
- [23] European Broadcasting Union. (2011). *EBU-TECH3342* [Online]. Available: <https://tech.ebu.ch/docs/tech/tech3342.pdf>.
- [24] E. Vickers, "The Non-flat and Continually Changing Frequency Response of Multiband Compressors," in *AES 129th Conv.*, San Francisco, CA, 2010.
- [25] P. Kraght, "Aliasing in Digital Clippers and Compressors," *J. Audio Eng. Soc.* 48 (11), Nov. 2000.
- [26] A. Travagliani, "Broadcast Loudness: Mixing, Monitoring and Control," in *122nd AES Conv.*, Vienna, Austria, 2007.
- [27] E. Skovenborg and T. Lund, "Loudness descriptors to characterize programs and music tracks," in *AES 125th Conv.*, San Francisco, CA, 2008.
- [28] J. Bergstra *et al.*, "Predicting genre labels for artists using FreeDB," *Proc. of the 7th Int. Conf. on Music Information Retrieval*, 2006, pp. 85-88.
- [29] Y. Hu and M. Ogihara, "Nexttone Player: A Music Recommendation System Based On User Behavior," *Proc. of the 12th Int. Conf. on Music Information Retrieval*, 2011, pp. 103-108.
- [30] A. Schindler *et al.*, "Facilitating Comprehensive Benchmarking Experiments On The Million Song Dataset," *Proc. of the 13th Int. Conf. on Music Information Retrieval*, 2012, pp. 469-474.
- [31] J.Samson. (2001). *Grove Music Online: Genre* [Online]. Available: <http://www.oxfordmusic.com/subscriber/article/grove/music/40599>.
- [32] I. Shepherd. (2014). *What is the Loudness War?* [Online]. Available: <http://dynamicrangeday.co.uk/about>.
- [33] T. Woodhead. (2015). *What is Mastering?* [Online]. Available: <http://www.hippocraticmastering.com/whatismastering.html>.
- [34] AllMusic. (2015). *Euro-Pop* [Online]. Available: <http://www.allmusic.com/style/euro-pop-ma0000004446>.
- [35] AllMusic. (2015). *Trip-Hop* [Online]. Available: <http://www.allmusic.com/style/trip-hop-ma0000002906>.
- [36] E. Skovenborg *et al.*, "Loudness Assessment of Music and Speech," in *AES 116th Conv.*, Berlin, Germany, 2004.
- [37] M. Walsh *et al.*, "Adaptive Dynamics Enhancement," in *AES 130th Conv.*, London, UK, 2011.
- [38] H. Robjohns, "The End Of The Loudness War?" *Sound on Sound*, Feb. 2014.
- [39] B. Katz and R. Katz, *Mastering audio: the art and the science*. Focal Press, 2007.
- [40] A.v. Ruschkowski, "Loudness war," in *Systematic and comparative musicology : concepts, methods, findings*. Albrecht Schneider (ed.), pp. 213-230, 2008.
- [41] B. Katz. (2013). *The Loudness War Has Been Won: Press Release* [Online]. Available: <http://www.digido.com/forum/announcement/id-6.html>.
- [42] Z. Ma *et al.*, "Implementation of an intelligent equalization tool using Yule-Walker for music mixing and mastering," in *AES 134th Conv.*, Rome, Italy, 2013.
- [43] E. Perez-Gonzales and J.D. Reiss, "Improved control for selective minimization of masking using inter-channel dependency effects," *Proc. of the 11th Int. Conf. on Digital Audio Effects (DAFX-08)*, pp. 75-81, 2008.
- [44] E. Deruty *et al.*, "Human Rock Mixes Exhibit Tight Relations Between Spectrum And Loudness." *J. Audio Eng. Soc.*, 62 (10), Oct. 2014.

THEME AND VARIATION ENCODINGS WITH ROMAN NUMERALS (TAVERN): A NEW DATA SET FOR SYMBOLIC MUSIC ANALYSIS

Johanna Devaney, Claire Arthur, Nathaniel Condit-Schultz, and Kirsten Nisula

School of Music, Ohio State University, USA

{devaney.12, arthur.193, condit-schultz.1, nisula.1}@osu.edu

ABSTRACT

The Theme And Variation Encodings with Roman Numerals (TAVERN) dataset consists of 27 complete sets of theme and variations for piano composed between 1765 and 1810 by Mozart and Beethoven. In these theme and variation sets, comparable harmonic structures are realized in different ways. This facilitates an evaluation of the effectiveness of automatic analysis algorithms in generalizing across different musical textures. The pieces are encoded in standard `**kern` format, with analyses jointly encoded using an extension to `**kern`. The harmonic content of the music was analyzed with both Roman numerals and function labels in duplicate by two different expert analyzers. The pieces are divided into musical phrases, allowing for multiple-levels of automatic analysis, including chord labeling and phrase parsing. This paper describes the content of the dataset in detail, including the types of chords represented, and discusses the ways in which the analyzers sometimes disagreed on the lower-level harmonic content (the Roman numerals) while converging at similar high-level structures (the function of the chords within the phrase).

1. INTRODUCTION

There are a wealth of musical scores in digitized form currently available. While the vast majority exist as images, a combination of hand encoding of the visual data and advances in optical music recognition (OMR) technology have increased the amount of symbolic music data available. Unfortunately, most of this data is unlabeled, limiting its utility in developing predictive systems for analyzing symbolically represented music. Accurately segmenting and labeling symbolic music data requires a higher level of musical expertise than can be reasonably obtained through crowd-sourcing platforms, like Mechanical Turk¹. Even with expert-annotators, there is the challenge of ensuring

¹<http://www.mturk.com/>



© Johanna Devaney, Claire Arthur, Nathaniel Condit-Schultz, and Kirsten Nisula.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Johanna Devaney, Claire Arthur, Nathaniel Condit-Schultz, and Kirsten Nisula. "Theme And Variation Encodings with Roman Numerals (TAVERN): A new data set for symbolic music analysis", 16th International Society for Music Information Retrieval Conference, 2015.

that they all conform to the same conventions in labeling the data. In this regard, conforming to the analytic approach in a published textbook provides a measure of consistency for analyzing classical music.

This paper presents the Theme And Variation Encodings with Roman Numerals (TAVERN) dataset², a new dataset of segmented and analyzed symbolic classical music. TAVERN consists of 27 theme and variations sets by Mozart and Beethoven, segmented into phrases and analyzed in terms of both Roman numeral chord labels and chord function. All of the pieces were analyzed in duplicate by different PhD-level music theory students and both the notes and analyses were encoded in Humdrum-related formats [9]. The dataset focuses on pieces in theme and variation form where the underlying harmony remains relatively constant across variations, while rhythmic and textural aspects of the music change. The utility of theme and variations in symbolic music analysis has been demonstrated in the case of folk songs [27, 28] and for both harmony [8, 14] and melody [5] in classical themes and variations. This is the first such dataset, however, that includes harmonic and functional data, facilitating the development of algorithms of automatic symbolic chord recognition and symbolic similarity, through a deeper understanding of the impact of texture on both of these tasks. This paper begins with a survey of existing symbolic music datasets, both annotated and unannotated, before describing in detail the annotation process and the contents of the dataset.

2. EXISTING DATASETS

As noted above, there is a growing number of unannotated symbolic music datasets available, many items of which are available in several collections. The most popular in MIR research are those that are hand-encoded and, to a certain degree, curated. This includes the KernScores dataset [22], which has more than 100,000 files in `**kern` format [9] from a range of styles from folk [23] to classical. A number of the kern score pieces are available in other datasets, such as the music21 corpus [3], which contains files in MusicXML [6] and `**kern` format. The music21 corpus also includes the Yale Classical Archives Corpus [29], which contains almost 9000 pieces/movements divided into vertical slices. The Yale corpus is also part of the ELVIS database [1] along with the Josquin Research

²<http://getTAVERN.org>

Project³ and a number of smaller corpora of other Renaissance composers. While some datasets are focused on making printed versions of the musical scores available, they often supply symbolic data. For example, the Mutoptopia Project⁴ contains not only PDFs of the scores but also hand-encoded Lilypond⁵ and MIDI files. The Peach-note dataset [26] provides similar access to the Petrucci Music Library⁶ by running OMR on the scanned scores, which typically has a higher error rate than hand encoding. Researchers have also made use of publicly available Band in a Box lead sheets, e.g. [4], and MIDI files, e.g., [15].

There is a much smaller number of harmonically annotated datasets. Temperley encoded the analyses from the *Tonal Harmony* textbook by Kostka and Payne [11] for his work on key finding [24] and examined statistical properties of harmony [25]. These encodings have been used by other researchers for evaluating symbolic chord recognition systems [12, 18]. The note data and annotations are available both in a format Temperley defined as “note files”⁷ and as MIDI files (with the chord annotation inserted as lyrics).⁸ The KSN harmonic annotations [10] provide Roman numeral labels with duration and inversion information for the Real World Computing (RWC) dataset [7] and have been used for modeling pitch structures in polyphonic music [19].

3. ANALYTIC APPROACH

TAVERN comprises 27 sets of theme and variations, 10 by Mozart and 17 by Beethoven (listed in Table 1). The Beethoven set is nearly complete, with 18 of his 20 theme and variation sets included (Opus 35 was excluded because of the inclusion of a fugue in the piece and WoO 79 was excluded because it included only 5 variations, which was below our 6 variation minimum). The Mozart set is less complete: due to time and resource restrictions, we temporarily sampled variations across his career (leaving out K. 24, 54, 180, 264, 352, 460, 500). Going forward we plan to analyze and include these variations in the dataset once additional resources become available.

The pieces have been analyzed in duplicate by multiple expert-annotators using the hierarchical model of harmony defined in [13] that includes both Roman numeral and function labels, specifically a variant of functional analysis known as the ‘Phrase Model’. Section 3.1 provides some background on the ‘Phrase Model’ in general and Section 3.2 describes the annotation process.

3.1 Phrase Model

Phrases are complete musical statements built from an ordered presentation of three harmonic functions and ending with a cadence. One way of analyzing phrases is in

³<http://jrp.ccarh.org/>

⁴<http://www.mutopiaproject.org>

⁵<http://www.lilypond.org>

⁶<http://imslp.org>

⁷<http://theory.esm.rochester.edu/temperley/kp-stats/index.html>

⁸<http://www.cs.northwestern.edu/~pardo/kpcorpus.htm>

Composer	Piece	# Variations
Mozart	K.25	7
	K.179	12
	K.265	12
	K.353	12
	K.354	12
	K.398	6
	K.455	10
	K.501	12
	K.573	9
	K.613	8
	Beethoven	WoO 63
WoO 64		6
WoO 65		24
WoO 66		13
WoO 68		12
WoO 69		9
WoO 70		6
WoO 71		12
WoO 72		8
WoO 73		10
WoO 75		7
WoO 76		8
WoO 77		6
WoO 78		7
WoO 80		32
Opus 34		6
Opus 76	6	

Table 1. Summary of the sets of themes and variations in the data set.

terms of functions. The tonic function at the beginning of a phrase serves to establish the tonal centre, and at the end of a phrase to signal its return. The pre-dominant function prepares for the arrival of the dominant function, which sets up an opposition to tonic. The tension created by the movement to the dominant is ultimately resolved by a return to tonic. A phrase typically contains all three harmonic functions, but may contain just tonic and dominant. The cadences may close with the dominant function (termed a half cadence) or return to the tonic function (termed an authentic or deceptive cadence, depending on the chords used). Ideas about functional harmony can be found in Rameau [20], although the specification of the terms tonic, pre-dominant, and dominant were not defined until the late nineteenth century by Riemann’ [21]. We have included function labels in addition to the Roman numeral labels because we believe that they are essential in developing and testing hierarchical models of harmony. Since function harmony has some limitations for music outside of the Classical era, we focused this dataset on Mozart and early-mid career Beethoven pieces.

The ‘Phrase Model’ is a contemporary adaption of Riemann’s thinking and is defined in several textbooks. For the purposes of this project, we followed the specifics laid out in *The Complete Musician* by Steven Laitz [13]. Gen-

Wo070: Theme

Beethoven

I V7 I I6 ii6 V
 T T T T P D

Wo070: Variation 3

Beethoven

I V7 I I IV V
 T T T T P D

Figure 1. Example of a theme and variation from the dataset with harmonic analyses marked, note the similarity in the harmonic structure and the differences in the texture.

erally, the majority of I and iii chords (i and III in the minor mode) have a tonic function, although inversions of these chords may have other function, such as I_4^6 functioning as dominant, depending on their harmonic context. vi (or VI) chords may have either a tonic or pre-dominant function, while ii or IV (ii^0 or iv) chords are typically pre-dominant. V and vii⁰ chords are typically assigned a dominant function, except for when their inversions occur in passing or neighbor contexts with I or vi chords in a tonic function. An example of the ‘Phrase Model’ analytical approach is shown in Figure 1. In the Theme, the Roman numerals I-V⁷-I-I⁶ are assigned a tonic function, with the V⁷ in the first bar functioning as an ornamentation of the surrounding I chords, rather than having a dominant function. The ii⁶ chord has a pre-dominant function and the V chord has a dominant function. Since the phrase ends on the dominant function, rather than returning to the tonic function, it ends with a half cadence. The variation has a similar structure, with the first 2.5 measures having a tonic function, the second half of the third measure having a pre-dominant function (albeit with a IV chord instead of ii⁶ chord), and the fourth measure having a dominant function.

3.2 Annotators

The annotators are three PhD-level music theory students, who each have spent at least two years teaching the harmonic analysis technique described in Section 3.1 to undergraduate students within the same curricular framework. Thus the annotators are intimately familiar with the workings of Laitz’s version of the ‘Phrase Model’ and its ana-

lytic conventions, ensuring a common interpretation across the annotators on these conventions. Each of the theme and variations sets was analyzed by two annotators, with the annotators analyzing 18 theme and variations sets each. The annotators worked independently, dividing each of the themes and variations into phrases on their own and analyzing each phrase both in terms of Roman numerals and phrase-level function. In cases where there was disagreement between the annotators, a third annotator reviewed the analyses and sided with one interpretation. The adjudicated version of the analysis was then joined with the note data, as described in Section 4.1. On occasion, the analyzers would disagree on the Roman numerals while still agreeing about the function of the chords, an example of which is discussed in Section 4.2 We believe that points of disagreement between the trained annotators are an interesting source of information, particularly if chord recognition algorithms run into accuracy issues in the same situations, and so we are also releasing the individual annotations in addition to the adjudicated data.

4. DATASET DETAILS

4.1 Encoding Format

The musical scores of pieces were converted from publicly available MIDI files sourced online. The MIDI files were less error-prone than running OMR on printed scores of the pieces, but still required some manual correction. In the correction process, the MIDI files were first converted into **kern format after which the errors were hand-


```

!!!COM: Beethoven
!!!OTL: 6 Variationen uber "Nel cor
!!!piu non mi sento" von G. Paisiello
!!!Variation: Thema a
**func **harm **kern **kern
#M6/8 #M6/8 #M6/8 #M6/8
*G: *G: *G: *G:
*tb8 * * *
8D 8V 8r 8dd
=
=
4.T 4.I 8GL 4b
. . 8B .
. . 8aD 8b
4.T 4.V7 8DL 4a
. . 8F# .
. . 8cJ 8a
. . s2 s2
= =
2.T 2.I 8GL 4g
. . 8B .
. . 8dJ 8g
. . 8BL 4r
. . 8G .
. . 8DJ 8dd
= =
= =
4.T 4.I6 8BBL 4dd
. . 8D .
. . 8GJ 8g
4.P 4.i16 8CL 4ee
. . 8E .
. . 8AJ 8ee
= =
= =
4.D 4.V/V 8DL 4.a
. . 8F# .
. . 8AJ .
4D 4V 8DL 8r
. . 8F# 8r
*_ *_ *_ *_
    
```

Figure 2. Example of the encoding format for the theme in Figure 1. The leftmost column contains the function labels, the second one contains the harmonic labels, and the remaining columns contain the notes. Dots indicate that a label is continued from a previous row while elements of another spine change.

corrected in reference to public domain scores available in the Petrucci Music Library (namely 19th century publications from Breitkopf & Hartel [2, 16, 17]). In the correction process, ornamentation and grace notes were removed in order to simplify the data. In addition to pitch and duration information, ****kern** format allows for information about slurs and stem directions to be encoded. Where this information was encoded in the MIDI data, it was converted into the ****kern** data.

The analyses were encoded as separate spines and then joined with the ****kern** data. For the Roman numeral analysis the existing ****harm** representation⁹ was used. In this format, the labels are the same as standard Roman numeral labels except that the inversions are marked with the letters a (for first inversion, typically notated as $\overset{6}{4}$ for triads or $\overset{6}{5}$ for seventh chords), b (for second inversion, $\overset{6}{4}$ or $\overset{6}{3}$), and c (for third inversion, $\overset{6}{3}$) in order to maintain consistency for the number of character used to indicate inversions. We developed a new format, named ****func**, for the function encoding, which simply consists of the labels T (tonic), P (pre-dominant), and D (dominant). Thus each file consists of one ****func** spine, one ****harm** spine and a number of

****kern** spines, each of which corresponds to one staff in the piano score. An example of a file, corresponding to the upper scores in Figure 1, is shown in Figure 2. Each file in the dataset represents one phrase, with measure numbers marked in reference to the entire piece. This allows for the phrases across the corresponding theme and variations to be easily recombined into a single piece while at the same time providing an indication of where each phrase begins and ends. The files are readable by Humdrum, a MATLAB parser for the files is currently available on github¹⁰ and extensions to the music21 Humdrum parser will be available shortly. We have also generated audio versions of each file from the symbolic data via MIDI.

4.2 Theme and Variations Form

All 27 of theme and variations sets in TAVERN are in ‘sectional’ form, meaning that all of the themes and variations are tonally-closed distinct units. Within the sets, the harmony remains relatively constant across the theme and variations, while the theme’s melody is embellished in the variations. Additional musical interest is created through changes in rhythm, tempo, texture, key, and mode. There are some inconsistencies in the harmonies across related themes and variations, but these are typically substitutions of different chords with the same harmonic function. An example of this is present in Figure 1, where the ii^6 chord in the penultimate measure of the theme is substituted with a IV chord in the variation. However, ii^6 and IV share two common notes (the 4th and 6th scale degrees) and a common function (P), meaning that this substitution has very little harmonic impact.

In total, the dataset consists of 1060 phrases. Of these, 66 phrases occur as codas to isolated variations, so for these phrases there is no corresponding phrase in the related theme or variations. These have been included for the purposes of completeness. Of the 1060 phrases, 917 of the phrases are in the major mode, with the remaining 143 being in the minor mode. Seven different major and minor keys are occur in the dataset: A, B flat, C, D, E flat, F, G). Within the phrases there are 290 unique sonorities (counting each inversion as a separate sonority), this includes both diatonic chords and applied chords. A tally of the top 40 unique chords with the highest number of occurrences (at least 25) is shown in Figure 3, along with the number of times that each chord occurs in each function. In addition to highlighting the large number of chords that are annotated in the dataset, Figure 3 also demonstrates the utility of annotating function labels by showing that most of the chord inversions have two if not three possible functions (depending on the context in which they occur). This highlights the need for such labelled data in order to learn these contexts, rather than simply relying on rule-based systems.

The relatively large proportion of non-standard tonic chords with a tonic function in Figure 3 (e.g., ii, IV, V, vii⁶) are a result of ‘embedded phrases’ within the tonic function in some of phrases [13]. An example of this is shown in the ****comments** spine of Annotator Two’s analysis of

⁹ <http://www.humdrum.org/Humdrum/representations/harm.rep.html>

¹⁰ <https://github.com/jcdevaney/TAVERN>

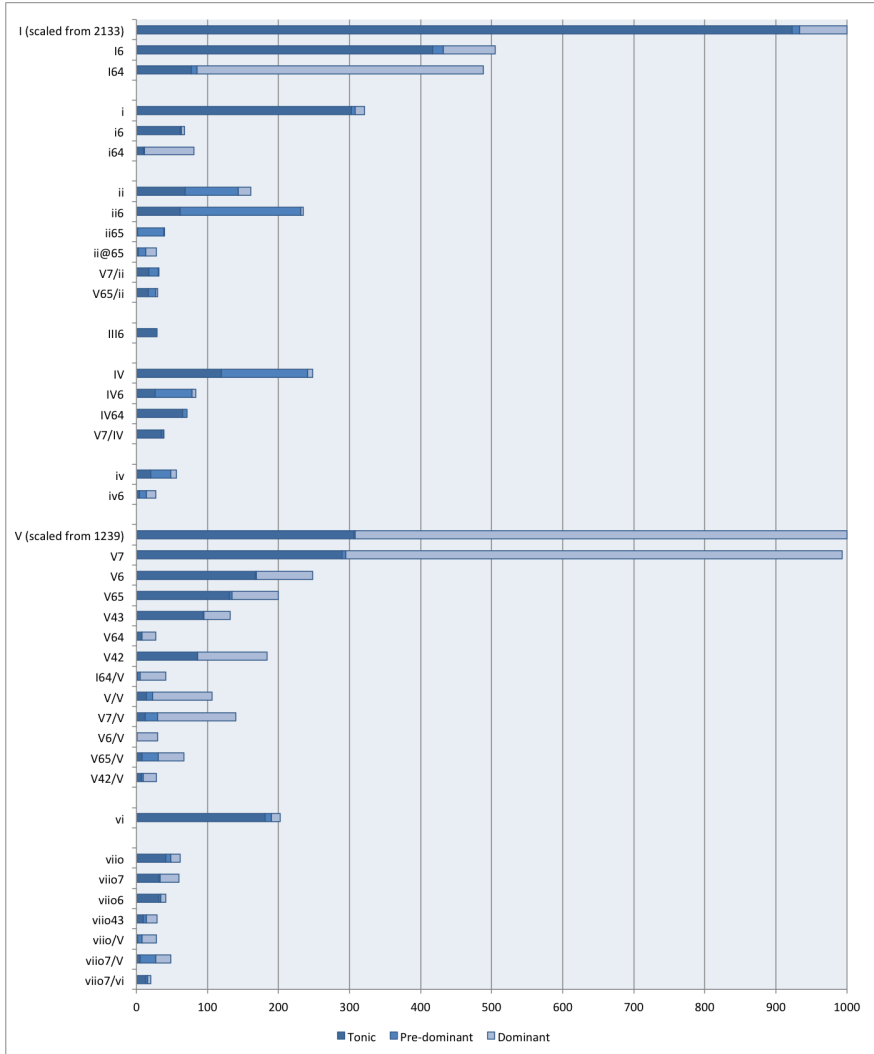
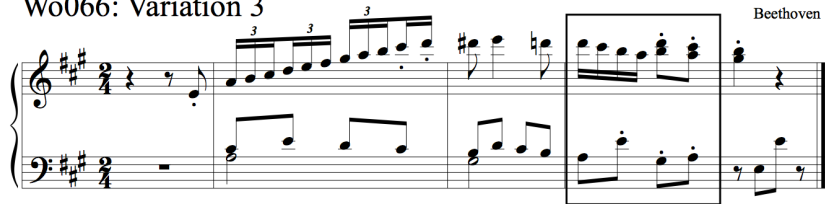


Figure 3. A tally of the number of times each of the top forty unique chords occurs in the dataset in regards to the function (Tonic, Pre-dominant, Dominant) in which they occur. The data for the I and V chords are shown the number of occurrences per 1000, scaled from their total number of occurrences (2133 and 1239 occurrences, respectively). This was done to facilitate the readability of the figure. The chords are grouped, from top to bottom, by the scale degree of their root note (or in the case of applied chords, the diatonic scale degree which functions as their relative tonic). Within each chord group, the chords are ordered by inversion followed by occurrences of applied dominant chords on that scale degree.

Wo066: Variation 3



Annotator 1

```

!!!COM: Beethoven
!!!OTL: Wo 066
!!!Variation: 3 a
**func **harm
*MZ/4 *MZ/4
*A: *A:
= =
8T 8V
= =
2T 2I
= =
2T 2V65
-----
2T 8I
. 8I64
. 8V65
. 8I
-----
4T 4.V
*_ *_
    
```

Annotator 2

```

!!!COM: Beethoven
!!!OTL: Wo 066
!!!Variation: 3 a
**func **harm **comments
*MZ/4 *MZ/4 *MZ/4
*A: *A: *A:
= = =
8T 8V
= = =
2T 2I T
= = =
2T 2V65 D
-----
2T 4I T
. 8vio6
. 8I
-----
4.D 4V
*_ *_ *_
    
```

Figure 4. An example of a phrase where the two annotators disagreed on specific chord labels. In the third measure (marked with a box), Annotator 1 analyzed the measure as ‘I- I₄⁶- V₅⁶-I’ while Annotator 2 analyzed the measure as ‘I-vii^{o6}-I’. The adjudicating annotator sided with Annotator 2 because in this context ‘vii^{o6}’ label is a complete chord. ‘V₅⁶’, despite being technically correct, is less desirable because the root of the chord (E) is missing. Annotator 2’s analysis also demonstrates the nomenclature of ‘embedded phrases’, which are marked when there is a low-level ‘T-P-D-T’ or ‘T-D-T’ pattern within the main T function that does not result in a cadence. Where applicable, ‘embedded phrase’ analyses are available in the individual annotators’ files in the ****comments** spine.

musical phrases reproduced in Figure 4. Instances of embedded phrases are not included in the main database files, but are available in the individual annotator’s files that are also released as part of TAVERN. Figure 4 also provides an example where the two annotators agreed on the overall harmonic function, but disagreed on the specific Roman numerals (as seen in the different analyses for measure 3). Ultimately, in this case, a third annotator determined the second annotator’s analysis to be superior both because the chord labels described complete chords and because it better mirrored the harmonic activity in the corresponding phrases in the related theme and variations.

5. CONCLUSIONS

This paper has presented TAVERN, a new dataset of 27 harmonically annotated theme and variations piano pieces by Mozart and Beethoven that will facilitate research on symbolic chord recognition and similarity in symbolic music. Each musical phrase in the dataset is encoded as a separate file. The note information is encoded in ****kern** for-

mat, the Roman numerals in ****harm** format, and the harmonic function of each Roman numeral label in the newly defined ****func** format.

This dataset will be useful for systematically evaluating the effect of textural changes on symbolic chord recognition algorithms since the consistency of harmonic materials and melodic frame across each theme and variations set occurs against a wide range of musical textures. Also, the segmentation of the pieces into phrases can facilitate the development and evaluation of algorithms for musical structure analysis. In addition to the symbolic music data, MIDI-generated audio files are available. In the future, we plan to use score-audio alignment to generate a mapping between the symbolic data and public-domain recordings of real piano performances, extending the utility of this dataset to include audio chord recognition research.

6. ACKNOWLEDGMENTS

This work was supported by the Google Faculty Research Award program.

7. REFERENCES

- [1] Christopher Antila and Julie Cumming. The vis framework: Analyzing counterpoint in large datasets. In *Proceedings of ISMIR*, pages 71–6, 2014.
- [2] Ludwig van Beethoven. *Variationen für das Pianoforte*, volume Serie 17 of *Ludwig van Beethovens Werke*. Breitkopf & Härtel, Leipzig, DE, 1862-90.
- [3] Michael Scott Cuthbert and Christopher Ariza. music21: A toolkit for computer-aided musicology and symbolic music data. In *Proceedings of ISMIR*, pages 637–42, 2010.
- [4] W. Bas De Haas, Martin Rohrmeier, Remco C Veltkamp, and Frans Wiering. Modeling harmonic similarity using a generative grammar of tonal harmony. In *Proceedings of the ISMIR*, pages 549–54, 2009.
- [5] Mathieu Giraud, Ken Déguernel, and Emiliós Cambouropoulos. Fragmentations with pitch, rhythm and parallelism constraints for variation matching. In *Sound, Music, and Motion*, pages 298–312. Springer, 2014.
- [6] Michael Good. MusicXML for notation and analysis. *The virtual score: representation, retrieval, restoration*, 12:113–24, 2001.
- [7] Masataka Goto, Hiroki Hashiguchi, Takuya Nishimoto, and Ryuichi Oka. Rwc music database: Popular, classical, and jazz music databases. In *Proceedings of ISMIR*, pages 287–288, 2002.
- [8] Keiji Hirata, Satoshi Tojo, and Masatoshi Hamanaka. Cognitive similarity grounded by tree distance from the analysis of k. 265/300e. In *Sound, Music, and Motion*, pages 589–605. Springer, 2014.
- [9] David Huron. *The Humdrum Toolkit: Reference Manual*. CCARH, Menlo Park, California, 1995.
- [10] Hitomi Kaneko, Daisuke Kawakami, and Shigeki Sagayama. Functional harmony annotation database for statistical music analysis. In *Proceedings of the ISMIR (Late Breaking Demo)*, 2010.
- [11] S. Kostka and D. Payne. *Tonal Harmony: With an Introduction to Twentieth Century Music*. McGraw-Hill, New York, NY, 2008.
- [12] Pedro Kröger, Alexandre Passos, Marcos Sampaio, and Givaldo De Cidra. Rameau: A system for automatic harmonic analysis. In *Proceedings of the International Computer Music Conference*, pages 273–281, 2008.
- [13] Steven G. Laitz. *The Complete Musician*. Oxford University Press, Oxford, 3rd edition, 2011.
- [14] Alan Marsden. Recognition of variations using automatic schenkerian reduction. In *Proceedings of ISMIR*, pages 501–506, 2010.
- [15] Matthias Mauch and Simon Dixon. A corpus-based study of rhythm patterns. In *Proceedings of ISMIR*, pages 163–168, 2012.
- [16] Wolfgang Amadeus Mozart. *Für ein und zwei Pianoforte zu vier Händen*. Wolfgang Amadeus Mozarts Werke, Serie XIX. Breitkopf & Härtel, Leipzig, DE, 1878.
- [17] Wolfgang Amadeus Mozart. *Variationen für das Pianoforte*. Wolfgang Amadeus Mozarts Werke, Serie XXI. Breitkopf & Härtel, Leipzig, DE, 1878.
- [18] B. Pardo and W. Birmingham. Algorithms for chordal analysis. *Computer Music Journal*, 26(2):27–49, 2002.
- [19] Stanislaw Andrzej Raczynski, Emmanuel Vincent, and Shigeki Sagayama. Dynamic bayesian networks for symbolic polyphonic pitch modeling. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(9):1830–1840, 2013.
- [20] Jean-Phillipe Rameau. *Treatise on Harmony*. Dover, Toronto, ON, 1722.
- [21] Hugo Riemann. *Harmony Simplified; or, the Theory of the Tonal Functions of Chords*. Augener, London, 1896.
- [22] Craig Stuart Sapp. Online database of scores in the humdrum file format. In *Proceedings of ISMIR*, pages 664–665, 2005.
- [23] Helmut Schaffrath and David Huron. *The Essen folk-song collection in the humdrum kern format*. CCARH, Menlo Park, CA, 1995.
- [24] David Temperley. *A Bayesian Approach to Key-Finding*, volume 2445 of *Lecture Notes in Computer Science*, pages 195–206. Springer Berlin Heidelberg, 2002.
- [25] David Temperley. A unified probabilistic model for polyphonic music analysis. *Journal of New Music Research*, 38(1):3–18, 2009.
- [26] Vladimir Viro. Peachnote: Music score search and analysis platform. In *Proceedings of ISMIR*, pages 359–362, 2011.
- [27] Anja Volk, WB Haas, and P Kranenburg. Towards modelling variation in music as foundation for similarity. In *Proceedings of the International Conference on Music Perception and Cognition*, pages 1085–1094, 2012.
- [28] Anja Volk and Peter van Kranenburg. Melodic similarity among folk songs: An annotation study on similarity-based categorization in music. *Musicae Scientiae*, 16(3):317–339, 2012.
- [29] Christopher White and Ian Quinn. The Yale-Classical Archives Corpus. In *Proceedings of the International Conference on Music Perception and Cognition*, page 320, 2014.

BENFORD'S LAW FOR MUSIC ANALYSIS

Isabel Barbancho¹ Lorenzo J. Tardón¹ Ana M. Barbancho¹
Mateu Sbert²

¹ Universidad de Málaga, ATIC Research Group, ETSI Telecomunicación,
Dpt. Ingeniería de Comunicaciones, Campus Teatinos, 29071 Málaga, Spain

² University of Girona, Institute of Informatics and Applications,
Campus Montilivi, 17003 Girona, Spain

ibp@ic.uma.es, lorenzo@ic.uma.es, abp@ic.uma.es, mateusbert@mac.com

ABSTRACT

Benford's law defines a peculiar distribution of the leading digits of a set of numbers. The behavior is logarithmic, with the leading digit 1 reflecting largest probability of occurrence and the remaining ones showing decreasing probabilities of appearance following a logarithmic trend. Many discussions have been carried out about the application of Benford's law to many different fields. In this paper, a novel exploitation of Benford's law for the analysis of audio signals is proposed. Three new audio features based on the evaluation of the degree of agreement of a certain audio dataset to Benford's law are presented. These new proposed features are successfully tested in two concrete audio tasks: the detection of artificially assembled chords and the estimation of the quality of the MIDI conversions.

1. INTRODUCTION

Benford's law, also known as the 'first-digit law', describes a peculiar distribution of the leading digits of datasets of numbers, especially those related to the measure of 'real-life phenomena'. Unlike the *central limit theorem*, Benford's law states that the typical distribution of the leading digits of a large number of datasets, derived from the measure of several common variables follows a logarithm-shaped law.

Most of the measures from real-life (tax returns, street addresses, population number or length of rivers) seem to present this peculiar distribution. Many works have been published on Benford's law, mixing the empirical evidence with some more mathematical formalism.

Benford's law has been widely proposed as a discriminating tool for 'naturally-shaped' datasets [6] and even employed [8] or criticized [5] as a somewhat reliable diagnostic tool to detect a large variety of frauds.

In this paper, Benford's law is evaluated as a discriminator for audio signals. In particular it is employed to detect

differences between natural and artificially created chords and real music and MIDI-generated music.

The article is organized as follows: in Section 2, Benford's law is discussed and its probabilistic framework is detailed. In Section 3, the three new audio features based on the evaluation of the degree of agreement of a certain dataset to Benford's law are defined. These descriptors are widely employed in Section 4 for the aforementioned tasks, as part of the audio signals analysis. Finally, in Section 5, some conclusions are drawn.

2. BENFORD'S LAW

Benford's law affirms that the frequency of occurrence of the leading significant digit of a large dataset coming from real-life measurements, presents a peculiar histogram in which the height of the bars follows a logarithmic scale (see Figure 1).

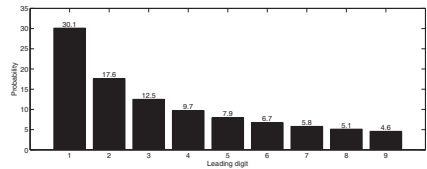


Figure 1. The logarithm-shaped distribution of the leading digits, following Benford's law.

More specifically, the probability value of the d -th digit is computed as follows:

$$P(d) = \log_{10} \left(1 + \frac{1}{d} \right) \quad (1)$$

where d is the digit number.

Simon Newcomb [11] first described this peculiar behavior after the observation of the pages of the tables of common logarithms. He noticed that the logarithms beginning with the digit 1 were more frequently browsed than the others. In his two-page paper, he briefly described the empirical evidence of such observation, extending it to all the digits.

© Isabel Barbancho, Lorenzo J. Tardón, Ana M. Barbancho, Mateu Sbert. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Isabel Barbancho, Lorenzo J. Tardón, Ana M. Barbancho, Mateu Sbert. "Benford's Law for Music Analysis", 16th International Society for Music Information Retrieval Conference, 2015.

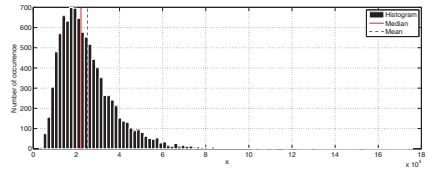
However, his work remained unknown for several years. In 1938, the *General Electric* physicist Frank Benford, apparently unaware of Newcomb's paper, formalized the same observations with a more consistent article published by the American Philosophical Society [4]. He included the formalization of the same law and a large amount of observations of real-life phenomena gathered during several years of research.

The rigorous mathematical discussion of the law was tackled several years after, and it is currently a matter of question. In 1976, the mathematician Ralph Raimi wrote about the mathematical explanation of the law, citing the 'scale-invariance' as one of the possible keys for interpretation of the phenomenon [14]. Theodore Hill [7], in 1995, described the statistical derivation of the law, while in 1997, Stephen Smith [15], in his book "The Scientist and Engineer's Guide to Digital Signal Processing" presented a rigorous complete description, under the point of view of the signal processing.

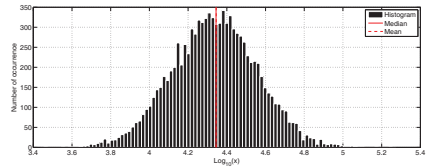
Nowadays, Benford's law is a well defined probabilistic problem, and it has been demonstrated that it is based on an intrinsic property of a large number of real-life datasets. According to the central limit theorem [13], the distribution of a certain measure of a quantity follows a normal distribution. The larger the amount of data, the closer the fit of the sample histogram to the Gaussian distribution. Nevertheless, when a single measure is iteratively repeated, its variance tends to be steady and to robustly define the range of variability of the quantity measured. Usually, it limits the width of the distribution to few orders of magnitude. In fact, it is infrequent that a series of iterative measures of the same variable could span across a wide range of values.

Also, if we multiply groups of random numbers, each following a normal distribution, we will obtain a new dataset following the so called 'log-normal' distribution [9]. Its name derives from the dome-shaped histogram that this kind of distribution shows, when it is represented on a logarithmic scale. In log-normal distributions, 95% of the values are distributed within the mean μ minus twice the standard deviation σ and the mean μ plus twice the standard deviation σ , on the logarithmic scale. This leads to an accumulation of values on the left edge of the distribution, on the linear scale [15]. Actually, in log-normal distributions the median is lower than the mean and they present large positive values of skewness [9] (see Figure 2).

The fact that the log-normal distribution usually derives from the combination of normally distributed variables, leads one to assume that, in nature, it is as common as the normal distribution [15]. Most of the datasets of real-life variables are log-normally distributed, especially those with only-positive values, where the intrinsic limitation leads to an increase of probability around the smallest values. Most of these datasets follow Benford's law. In environmental pollutants datasets, for instance, most of the measures are typically very low and only few of them are larger than their mean. Moreover, these variables are typically only positive, but they usually show very low values, very close to zero. This leads to a compression of the



(a) The histogram of a log-normal shaped dataset (linear axis).



(b) The histogram of a log-normal shaped dataset (logarithmic axis).

Figure 2. An example of log-normal shaped dataset in linear and logarithmic axis. The median and the mean are represented with continuous and dashed line, respectively. Note that the median and the mean coincide when the histogram is spaced on the logarithmic axis (the distribution is normally-shaped). The histogram bins have been equally spaced on the logarithmic axis, such to define a constant width of the bars.

histogram toward the minimum, resulting in a typical log-normal distribution.

Nevertheless, the shape of the histogram is not sufficient to be an index of the degree of fit to Benford's law. Usually, the log-normal distributions derived from the combination of multiple normal distributions (with different widths) are broader than them, because of the larger range of variability they present. In fact, it is the width of these kinds of distributions, that is key to understand their relation to Benford's law. Smith [15] shows how the degree of fit to the law of a certain dataset is a mere question of distribution width. The broader the distribution of the data, the more accurate the fit to the theoretical law.

This is a very important issue, related to the data manipulation by humans. The most common way to systematically extract the leading digit of a number is to multiply or divide it by ten, until it reaches a value between 1 and 9.9 periodic. In particular, the number must be divided by 10, if it is higher than 10 and multiplied by 10, if it is lower than 1.

Thus, for instance, the number 0.00567 will be multiplied by 10 three times to obtain the number 5.67, whose integer part (5) is taken into account as the leading digit. Similarly, for the number 7865, it has to be divided by 10 three times to obtain 7.865, and the correspondent leading digit (7).

This 'human-driven' mechanism is primarily responsible for dependence of the distribution of the leading digits

on the logarithmic law [15]. Hence, the amount of dependence, namely the degree of fit to Benford's law, depends on the broadness of the original data distribution. If the data span across a large number of orders of magnitudes, with respect to unity, they will need several steps of multiplications/divisions to be scaled to range between 1 and 9.9 periodic. Conversely, if the dataset ranges from 1 to 9, the numbers will not require any operation. The impact of these kinds of manipulations is directly related to the degree of agreement to Benford's law.

3. BENFORD'S LAW BASED AUDIO FEATURES

In order to evaluate the degree of agreement of a certain dataset to Benford's law, several approaches can be employed. The task is to obtain new features to be used as comparative measure among the different audio elements to be classified. In this section three new features will be extracted: the one-scaling-test, the Fourier-based method and the goodness-of-fit test.

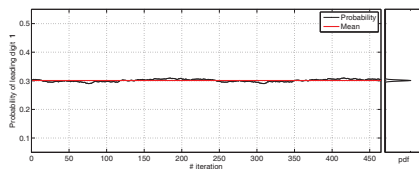
3.1 The one-scaling-test

Raimi [14], still speaking about a "universal law", introduced the scale-invariance principle to define the validity of the law. He affirmed that "...since God is not known to favor either the metric system or the English system...", Benford's law must be scale-invariant. Smith [15] formalized a test based on the scale-invariance of the law, by measuring the variation of the probability of occurrence of the leading digit 1, when the dataset is iteratively multiplied by a constant.

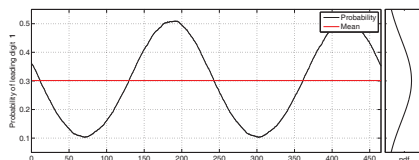
The theoretical probability of occurrence, by Benford's law, of the first leading digit is 0.301. If the empirical probability of the digit 1 of a certain dataset is close to this value, we can suppose that the dataset follows (potentially) Benford's law. This is obviously not sufficient. Recalling the concept about the scale-invariance, we can affirm that if the dataset follows the law, the empirical probability of occurrence of its leading digits (the digit 1 in this case) should not vary, or vary very weakly, if the dataset is iteratively multiplied or divided. The so called *one-scaling-test* proposed by Smith [15] exploits this property to evaluate the agreement of a certain dataset to Benford's law.

If we take two log-normally distributed datasets with equal mean, 10, and different standard deviation, 0.5 and 3, respectively, and we multiply them iteratively by a constant (e.g.: 1.01), we will observe a certain variation of the probability of occurrence of the first leading digit around the value 0.301 (Figure 3).

A broader distribution presents a much weaker variation of the probability of occurrence of the first leading digit around the value 0.301, than a narrower distribution. The means of the distribution of the probability values are 0.3010 and 0.3013, for the broader and the narrower distribution, respectively. That is, they both follow Benford's law, showing a value close to the expected theoretical probability (0.301). However, their standard deviations 0.0069 and 0.1450 reveal a much larger variation around the mean



(a) The variability of the probability of occurrence of the leading digit 1 for a broad log-normally distributed dataset ($\sigma = 3$).



(b) The variability of the probability of occurrence of the leading digit 1 for a narrow log-normally distributed dataset ($\sigma = 0.5$).

Figure 3. An example of the effect of the width of the (log-normal) distribution on the one-scaling-test. Means are represented with thick lighter line. The equivalent PDFs are displayed on the right side of the plots.

for the dataset with the narrower distribution. Although both datasets seem to follow Benford's law, the broader one required a heavier manipulation of the original data to extract the leading digits and it emphasized the logarithmic pattern attributed to their distribution, leading it to approach the theoretical law closer.

Note that in both cases, the variation of the probability shows a periodic pattern due to the factor chosen for the multiplication. The leading digit is unchanged when the numbers are multiplied by 10. In our example, this occurs every 232 times ($1.01^{232} \approx 10$).

The one-scaling-test presents a main drawback related to the high computational cost derived from the iterative multiplication of the whole dataset. If we consider one single minute of an audio signal recorded at a sampling frequency of 44.1 kHz, we have to handle with a vector of more than 2.5 millions samples. If we want to multiply this dataset at least $232 + 1$ times (to observe at least one whole period), we must do more than 600 millions of operations. In the case of exploiting Benford's law in a classifier tool for music genres, we should have to handle hundreds of songs, each of them with a length of several minutes. This would become an unfeasible task from the point of view of the computational cost.

3.2 The Fourier transform-based method

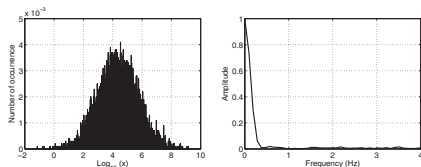
Smith [15] reinterprets the problem from the point of view of signal processing. He proves that the degree of agreement of a certain dataset to Benford's law, can be estimated by evaluating the behavior of the Fourier transform (FT) of the normalized histogram in logarithmic axes. In particu-

lar, the measure of how fast the transform falls, from its maximum value (1 at frequency 0) to its minimum value (zero at some frequency higher than zero), is directly related to the width of the distribution measured with the normalized histogram and, consequently, with the degree of correspondence with the law.

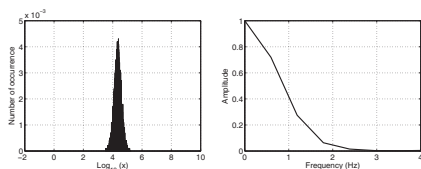
Ideally, in order to follow perfectly Benford's law, the Fourier transform should present a unitary value at frequency zero and a zero value at all the remaining frequencies. This would occur if the distribution was uniform from $-\infty$ to $+\infty$ [12].

In real-life, this does not occur. Hence, the faster the Fourier transform drops to zero, the closer the agreement of the dataset to Benford's law. In particular, Smith defines the value at frequency 1 as the threshold to discriminate between the agreement or not of the dataset to the law [15]. If the transform of the histogram in logarithmic axes (denoted as PDF) falls to zero before frequency $1Hz$, the correspondent dataset follows Benford's law. If it does not occur, the dataset does not follow the law. In practice, the value of the PDF at $f = 1Hz$ is a reliable index of the degree of agreement with the law.

In Figure 4, an example of the application of the Fourier transform to the histograms of the dataset tested in the previous section, is shown.



(a) Broad log-normally distributed dataset. Left: distribution on a logarithmic axis. Right: Fourier transform of the distribution (PDF).



(b) Narrow log-normally distributed dataset. Left: distribution on a logarithmic axis. Right: Fourier transform of the distribution (PDF).

Figure 4. Example of the application of the Fourier transform for the estimation of the agreement of the data to Benford's law. The transform of the broader distribution drops to zero faster than the narrower one, revealing a closer correlation with the law.

The distribution of the data shown in Figure 4(a) is broader than the one in Figure 4(b). Actually the two datasets are the same that were previously analyzed in Figure 3, with standard deviation 3 and 0.5, respectively. The PDF of the broader distribution falls to zero much faster than the narrower one. In particular, the amplitude of the PDF at frequency $1Hz$ is 0.0023 and 0.4184, for the broader

and the narrower distribution, respectively. This issue reveals a closer agreement to Benford's law of the broader distribution, as observed previously.

Note that unlike the one-scaling-test, the method based on the Fourier transform has a reasonable computational cost. Furthermore, this method returns a higher discriminant range for the two datasets: The ratio between the two standard deviations of the one-scaling-test is about 20, while the ratio between the two values of the transforms at frequency $1Hz$ is about 180. If the aim of the application of Benford's law is a boolean discrimination of the data, then the Fourier transform-based method is efficient.

3.3 The χ^2 divergence and the goodness-of-fit test

An alternative to the two empirical methods proposed so far, is the well known χ^2 test [9]. It is called the *goodness-of-fit* test and it returns a measure of how well an empirical distribution fits a theoretical one.

The divergence is calculated as follows:

$$D = \frac{(f(d) - P(d))^2}{P(d)} \quad (2)$$

where $f(d)$ is the empirical relative frequency of the digit d and $P(d)$ stands for its theoretical probability defined, in our case, by Benford's law, detailed in equation (1).

The null hypothesis H_0 is verified if its associated probability (the p -value) does not exceed the significance level fixed a priori. This probability value, when the test is passed, can be employed as additional information for the measure of the agreement to Benford's law.

The goodness-of-fit test applied to the two datasets analyzed before, returns a divergence value of 0.2484 and 0.0009, for the narrower and the broader distribution, respectively. Actually, the narrower distributed dataset did not pass the test. In Figure 5, the two empirical distributions of the leading digits are shown.

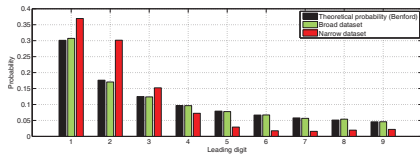


Figure 5. The distribution of the leading digits for the two datasets of the previous example. Both empirical distributions are compared against the theoretical values.

Once again, the broader distribution reveals a larger correlation with Benford's law, than the narrower one. Note that the ratio between the two divergences (about 280) is even larger than the one measured between the two values of the transform in the previous example.

Nevertheless, the approach based on the Fourier transform does not need to extract the samples in the dataset and it is, therefore, more efficient.

4. EVALUATION OF BENFORD'S LAW PERFORMANCE IN PATTERN RECOGNITION TASKS FOR MUSIC SIGNALS

In this section, the performance of the proposed features based on quantitative measurements of agreement to Benford's law is evaluated in two concrete audio tasks.

4.1 Real and artificially assembled chords

Often, the methods employed in the evaluation of the algorithms for multi-pitch estimation are based on the usage of ground-truth datasets of artificially assembled chords, i.e. made up by the summation of individual waveforms of the single notes that compose the chords. In this context, this procedure leads to cleaner spectra that can be more easily analysed. Benford's law based audio features are employed to discriminate between real and artificially assembled chords. A set of 230 chords has been examined. The half of them are real chord [3] and the other half are the same chords but artificially assembled adding single notes. The two sets of chords do not reveal any kind of significant difference when submitted to a perceptual evaluation. They sound practically the same.

Using this data set, the descriptors to evaluate the agreement of the data to Benford's law have been calculated for each pair of chords (real and artificially assembled). The ones-scaling test has not been performed because of its high computational costs. In order to evaluate the performance of the new Benford's law based features, they have been compared against a set of time and frequency features commonly used for the music classification task (RMS, ZCR, CER, SPF) [16].

The descriptors defined in this context reflect a notable discrepancy between the two classes of chords. Surprisingly, an average value of the 30% of the samples (12 out of 115 for the artificial chords and 56 out of 115 of the real chords) did not pass the χ^2 test. The signals showed rather skewed distributions on the logarithmic axis with the consequent decrease of the level of agreement to Benford's law.

A *knn* classifier has been adopted here to perform the classification of the chords using both the set of single features selected and the two groups of features with and without the two Benford's law-based descriptors. As it is shown in Table 1, Benford's law-based features behave rather well when compared against the typical features for audio classification. Also, the multidimensional set of descriptors improves its performance with the inclusion of the two Benford's law-based features. It is interesting to note that the artificial chords returned smaller values of $PDF(f = 1\text{Hz})$ than the real chords (see Figure 6).

4.2 Quality of MIDI conversion

Recalling the 'Nature-dependence' of Benford's law, we formulate the hypothesis that the agreement of MIDI [10] audio to Benford's law, could be used as a ranking measure for the quality of automatic MIDI converters. Two software tools for automatic MIDI conversion were tested: the

Feature	Success rate (%)
Benford's law-based features	
$PDF(f = 1\text{Hz})$	80.87
χ^2 divergence	71.74
Time and Frequency features	
Root mean square	82.61
Zero crossing rate	68.70
Cepstrum residuals	58.26
Spectral flux	72.61
Grouped-feature set	
Time and Frequency features set	79.57
Benford's law-based features added	82.17

Table 1. Real and artificial chord classification accuracy of the single-feature tests and the grouped-feature tests.

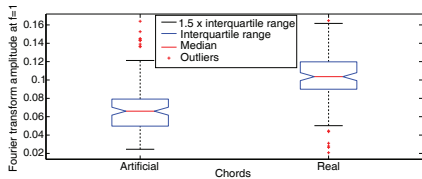


Figure 6. Box-whiskers plot of the amplitude of $PDF(f = 1\text{Hz})$, for artificial and real chords. The non-overlapping notches, indicating the 95% confidence interval of the two medians, reveal good discrimination power of the analyzed feature.

freeware software AMAZING MIDI (v1.70) by arakisoftware [2] and the shareware software AKoff Music Composer (v2.0) by the AKoff Sound Labs [1]. Both of them present at least two different configuration sets. In particular, the AKoff software has been run with and without the application of the 'overtones filtering', a utility to filter the highest harmonics of the spectrum, while the AMAZING MIDI software has been executed with and without a time and an amplitude filter (to reduce the range of amplitude and note duration).

The term "quality of a MIDI conversion" is a rather subjective concept, i.e., it may depend on the person who is evaluating that quality. Therefore, the sounds of the automatic conversion tools have been listened carefully by a team of ten expert musicians who have evaluated personally both the similarity between the converted track and the original one, and the overall quality of the MIDI audio. Each listener had to rank the MIDI converters with a score in the range 0 (the worst quality) to 100 (the best quality). Table 2 shows the mean of the subjective test scores obtained by each tool/configuration.

In Figure 7, an example of the test performed, applied to the song 'Come sei veramente' by the pianist G. Allevi, is shown. The original track returned the smallest value in the ones-scaling test, $PDF(f = 1\text{Hz})$ and the χ^2 diver-

Software/configuration	Mean score
AKoff with overtone control	27/100
AKoff without overtone control	48/100
AmazingMIDI with filters	75/100
AmazingMIDI without filters	80/100

Table 2. Mean subjective ranks of the four combination of tool and configuration employed in the MIDI-quality test.

gence, with respect to the other four MIDI versions. The two outcomes of the AKoff software returned the largest values of each descriptor, revealing the lowest accordance to Benford's law. Note the relation between features extracted and the subjective ranks in Table 2. Therefore, the accordance to Benford's law provide us with a measure of the quality of the MIDI converters.

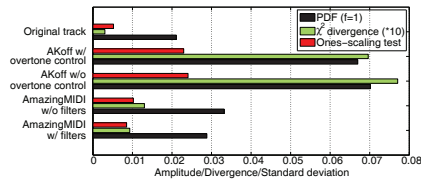


Figure 7. The three Benford's law-based features calculated for the track 'Come sei veramente' by the Italian pianist Giovanni Allevi. Divergence values are multiplied by 10 for displaying purposes.

5. CONCLUSIONS

In this paper, it has been shown how Benford's law can be conveniently exploited to extract useful features that can be successfully used in different audio pattern recognition tasks. Three new Benford's law based audio features based on different measurement of the agreement to Benford's law have been proposed.

Two concrete tasks have been addressed to highlight this novel context of application of Benford's law for audio signal. For chord analysis, the new proposed features are rather compelling as good discriminators when compared against other typical features for speech and audio classification and also the results obtained for the determination of the quality of the automatic MIDI conversions are promising.

Therefore, through this paper it has been illustrated how Benford's law, that substantially arises as a matter of shape and width of the distribution of the leading digits of the data, can be conveniently exploited for audio classification problems.

6. ACKNOWLEDGEMENT

This work has been funded by the Ministerio de Economía y Competitividad of the Spanish Government under Project No. TIN2013-47276-C6-2-R and Project No. TIN2013-47276-C6-1-R. This work has been partially done at Universidad de Málaga, Campus de Excelencia Internacional (CEI) Andalucía Tech.

7. REFERENCES

- [1] AKoff-Sound-Labs. AKoff music composer, wav-to-midi converter. <http://www.akoff.com/music-composer.html>, 2011.
- [2] Arakisoftware. AmazingMIDI, wav to midi converter for music transcription. <http://www.pluto.dti.ne.jp/~araki/amazingmidi>, 2011.
- [3] Ana M. Barbancho, Isabel Barbancho, Lorenzo J. Tardón, and Emilio Molina. *Database of Piano Chords. An Engineering View of Harmony*. Springer-Verlag, New York, NY, USA, 2013.
- [4] Frank Benford. The law of anomalous numbers. *Proceedings of the American Philosophical Society*, 78(4):551–572, 1938.
- [5] Joseph Deckert, Mikhail Myagkov, and Peter C. Ordeshook. The irrelevance of benfords law for detecting fraud in elections. *Caltech/MIT Voting Technology Project*, 1(9), 2010.
- [6] Esteve del Acebo and Mateu Sbert. Benford's law for natural and synthetic images. In *Computational Aesthetics*, pages 169–176. Eurographics Association, 2005.
- [7] Theodore P. Hill. A statistical derivation of the significant-digit law. *Statistical Science*, 10(4):354–363, 1995.
- [8] Brian A. Jacob and Steven D. Levitt. Rotten apples: An investigation of the prevalence and predictors of teacher cheating. *The Quarterly Journal of Economics*, 118(3):843–877, 2003.
- [9] Averill M. Law and W. David Kelton. *Simulation Modeling and Analysis*. McGraw Hill, Inc., 1991.
- [10] Robert A. Moog. Midi: Musical instrument digital interface. *Journal of the Audio Engineering Society*, 34(5):394–404, 1986.
- [11] Simon Newcomb. Note on the frequency of use of the different digits in natural numbers. *American Journal of Mathematics*, 4(1):39–40, 1881.
- [12] Alan V. Oppenheim and Ronald W. Schaffer. *Discrete-Time Signal Processing, 3/E*. Prentice Hall, 2010.
- [13] Athanasios Papoulis and S. Unnikrishna Pillai. *Probability, Random Variables and Stochastic Processes*. McGraw Hill, 2002.

-
- [14] Ralph A. Raimi. The first digit problem. *The American Mathematical Monthly*, 83(7):521–538, 1976.
 - [15] Steven W. Smith. *The scientist and engineer's guide to digital signal processing*. California Technical Publishing, San Diego, CA, USA, 1997.
 - [16] Lorenzo J. Tardón, Simone Sammartino, and Isabel Barbancho. Design of an efficient music-speech discriminator. *The Journal of the Acoustical Society of America*, 127:271–279, 2010.

AN AUDIO TO SCORE ALIGNMENT FRAMEWORK USING SPECTRAL FACTORIZATION AND DYNAMIC TIME WARPING

J.J. Carabias-Orti¹ F.J. Rodriguez-Serrano² P. Vera-Candeas²
N. Ruiz-Reyes² F.J. Cañadas-Quesada²

¹ Music Technology Group (MTG), Universitat Pompeu Fabra, Spain

² Polytechnical School of Linares, Universidad de Jaen, Spain

julio.carabias@upf.edu

ABSTRACT

In this paper, we present an audio to score alignment framework based on spectral factorization and online Dynamic Time Warping (DTW). The proposed framework has two separated stages: preprocessing and alignment. In the first stage, we use Non-negative Matrix Factorization (NMF) to learn spectral patterns (i.e. basis functions) associated to each combination of concurrent notes in the score. In the second stage, a low latency signal decomposition method with fixed spectral patterns per combination of notes is used over the magnitude spectrogram of the input signal resulting in a divergence matrix that can be interpreted as the cost of the matching for each combination of notes at each frame. Finally, a Dynamic Time Warping (DTW) approach has been used to find the path with the minimum cost and then determine the relation between the performance and the musical score times. Our framework have been evaluated using a dataset of baroque-era pieces and compared to other systems, yielding solid results and performance.

1. INTRODUCTION

In this work, we address the problem of audio-to-score alignment (or score matching), which is the task of synchronizing an audio recording of a musical piece with the corresponding symbolic score. There are two approaches to this problem, often called “offline” and “online” alignment. In offline alignment, the whole performance is accessible for the alignment process, i.e. it allows us to “look into the future” while establishing the matching. This is interesting for applications that do not require the real-time property such as Query-by-Humming, intelligent audio editors and as a front-end for many music information retrieval (MIR) systems. Online alignment, also known as score following, processes the data in realtime as the

signal is acquired. This tracking is very useful for applications such as automatic page turning, automated computer accompaniment of a live soloist, synchronization of live sound processing algorithms for instrumental electroacoustic composition or the control of visual effects synchronized with the music (e.g. stage lights or opera super-titles).

Audio-to-score alignment is traditionally performed in two steps: feature extraction and alignment. On the one hand, the features extracted from the audio signal characterize some specific information about the musical content. Different representations of the audio frame have been used such as the output of a short-time Fourier transform (STFT) [1], auditory filter bank responses [2], chroma or “chroma-like” vectors [3, 4], multi-pitch analysis information [5–8]. On the other hand, the alignment is performed by finding the best match between the feature sequence and the score. In fact, most systems rely on cost measures between events in the score and in the performance. Two methods well known in speech recognition have been extensively used in the literature: statistical approaches (e.g. HMMs) [6–11], and dynamic time warping (DTW) [3, 12, 13].

In this paper we propose an audio to score framework based on two stages: preprocessing and alignment. On the first stage, we analyze the provided MIDI score to define the set of combinations of concurrent notes and the transitions between them (i.e. the different states of the provided MIDI). Then the score is converted into a reference audio signal using a synthesizer software and we use a method based on Non-Negative Matrix Factorization (NMF) with Beta-divergence to learn spectral patterns (i.e. basis functions) for each combination of notes. A similar approach was used by Fritsch and Plumbey in [14], but they use one component per instrument and note plus some extra-components to model the residual sounds. NMF was also used by Cont [8] as a multi-pitch estimator which defines the observation model. Joder et al. [10] also defined a set of template vectors for each combination of concurrent notes but directly from the score (i.e. without using audio synthesis). The combination templates are obtained as a linear mapping of individual notes trained patterns using several representations. On the second stage, alignment is performed in two steps. First, the matching measure between events in the score and in the performance is defined.



© J.J. Carabias-Orti, F.J. Rodriguez-Serrano, P. Vera-Candeas, N. Ruiz-Reyes, F.J. Cañadas-Quesada. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** J.J. Carabias-Orti, F.J. Rodriguez-Serrano, P. Vera-Candeas, N. Ruiz-Reyes, F.J. Cañadas-Quesada. “An Audio to Score Alignment Framework using Spectral Factorization and Dynamic Time Warping”, 16th International Society for Music Information Retrieval Conference, 2015.

Concretely, a divergence (i.e. cost) matrix is estimated using a low latency signal decomposition method previously developed by the authors in [15] that uses the spectral patterns fixed from the previous stage. Finally a DTW strategy has been used to find the path with the minimum cost and then determine the relation between the performance and the musical score times. Both, offline and online DTW approaches are implemented as in [23] and [13], respectively.

The structure of the rest of the paper is as follows. In Section 2, we briefly review the DTW principles. In Sections 3 the proposed audio to score framework is explained. In Section 4, the evaluation set-up is presented and, in Section 5 the proposed method has been tested and compared with other reference systems. Finally, we summarize the work and discuss future perspectives in Section 6.

2. DTW BACKGROUND

DTW is a technique for aligning two time series or sequences. The series are represented by 2 vectors of features $U = u_1, \dots, u_i, \dots, u_J$ and $V = v_1, \dots, v_j, \dots, v_J$ where i and j are the point indices in the time series. I and J represent the length of time series U and V , respectively. As a dynamic programming technique, it divides the problem into several sub-problems, each of which contribute in calculating the distance (or cost function) cumulatively.

The first stage in the DTW algorithm is to fill a local distance matrix (a.k.a cost matrix) D as follows:

$$D(i, j) = \psi(u_i, v_j) \quad (1)$$

where matrix D has $I \times J$ elements which represent the match cost between every two points in the time series. The cost function ψ could be any cost function that returns cost 0 for a perfect match, and a positive value otherwise (e.g. euclidean distance).

In the second stage (forward step), a warping matrix C is filled recursively as:

$$C(i, j) = \min \left\{ \begin{array}{l} C(i, j - c_j) + D(i, j) \\ C(i - c_i, j) + D(i, j) \\ C(i - c_i, j - c_j) + \sigma D(i, j) \end{array} \right\} \quad (2)$$

where c_i and c_j are step size at each dimension and range from 1 to α_i and 1 to α_j , respectively. α_i and α_j are the maximum step size at each dimension. Parameter σ controls the bias toward diagonal steps. $C(i, j)$ is the cost of the minimum cost path from $(1, 1)$ to (i, j) , and $C(1, 1) = D(1, 1)$.

Finally, in the last stage (traceback step), the minimum cost path $w = w_1, \dots, w_k, \dots, w_K$ is obtained by tracing the recursion backwards from $C(I, J)$. Each w_k is an ordered pair (i_k, j_k) such that $(i, j) \in w$ means that the points u_i and v_j are aligned. Moreover, the path has to satisfy the following three conditions: i) w is bounded by the ends of both sequences, ii) w is monotonic and iii) w is continuous.

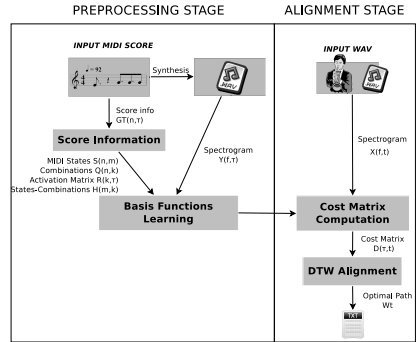


Figure 1: Block diagram of the proposed system

3. SYSTEM OVERVIEW

The proposed framework for audio-to-score alignment is presented in Figure 1. As can be seen, the framework has two stages. First, the preprocessing stage must be computed beforehand and only the MIDI score is required. Then, once the parameters are learned, alignment can be computed in realtime.

3.1 Preprocessing Stage

In this stage, the parameters for the alignment are learned from the score, which must be provided beforehand using MIDI representation. This stage is performed in two successive steps: states definition and spectral patterns learning, as detailed below.

3.1.1 States Definition

The aim of this step is to adequately organize the information given by the score to be used for alignment purposes.

First of all, the binary ground-truth transcription matrix $GT(n, \tau)$ (see Figure 2(a)) is inferred from the MIDI score, where τ is the time in frames referenced to the score (MIDI time) and n are the notes in MIDI scale. In Figure 2(a) the MIDI score involves just one instrument (a piano) but more instruments can be defined in a score. For those cases the n index refers to each note of the different instruments. Consequently, the number of total notes for a composition, N , is obtained as the sum of the number of different notes per instrument. The score can be interpreted as a consecutive sequence of M states. Each state m is defined by its combination of concurrent notes in the score (for all instruments). Also, the score informs about the time changes from one state to the next state. In fact, a score follower must determine the time (referenced to the input signal) of all transitions between states. There are only K unique combination of notes in a score where $K \leq M$ because some states represent the same combination of notes.

From the ground-truth transcription matrix $GT(n, \tau)$,

we obtain the following decomposition of binary matrixes

$$\mathbf{GT}(n, \tau) = \mathbf{Q}(n, k)\mathbf{R}(k, \tau) \quad (3)$$

where $\mathbf{Q}(n, k)$ is the notes-to-combination matrix, k the index of each unique combination of notes and $\mathbf{R}(k, \tau)$ represents the activation of each combination in MIDI time. In Figure 2(b), the note-to-combination matrix $\mathbf{Q}(n, k)$ is represented. This matrix contains the notes belonging to each combination but no information about MIDI time. Conversely, $\mathbf{R}(k, \tau)$ matrix retains the MIDI time activation per combination but no information about the notes active per combination, as can be seen in Figure 2(d).

In order to obtain the information for states required to perform the alignment, the notes-to-combination matrix $\mathbf{Q}(n, k)$ is further decomposed as

$$\mathbf{Q}(n, k) = \mathbf{S}(n, m)\mathbf{H}(m, k) \quad (4)$$

where $\mathbf{S}(n, m)$ is the notes-to-state matrix, m the index for the states, M the number of states and $\mathbf{H}(m, k)$ represents the unique combination k of notes active at each state m . In Figure 2(c), the notes-to-state matrix $\mathbf{S}(n, m)$ is represented, this matrix contains the notes belonging to each state, while $\mathbf{H}(m, k)$ matrix informs about the combinations active at each state, as can be seen in Figure 2(e).

The matrixes here defined will be used in the next stages to perform the alignment and are computed from the MIDI score.

3.1.2 Spectral Patterns Learning

When a signal frame is given to a score follower, the first step should be the computation of a similarity measure between the current frame and the different combinations of notes defined by the score. Our approach is to compute a distance (or divergence) between the frequency transform of the input and just one spectral pattern per combination of notes. A spectral pattern is here defined as a fixed spectrum which is learned from a signal with certain characteristics. The use of only one spectral pattern per combination allows us to compute the divergence with a low complexity signal decomposition method. This means that our method must learn in advance the spectral pattern associated to each unique combination of notes for the score. To this end, a state-of-the-art supervised method based on Non-Negative Matrix Factorization (NMF) with Beta-divergence and Multiplicative Update (MU) rules [15] is used, but in this work, we propose to apply it on synthetic signal generated from the MIDI score¹ instead of the real audio performance.

First of all, let us define the signal model as

$$\mathbf{Y}(f, \tau) \approx \hat{\mathbf{Y}}(f, \tau) = \mathbf{B}(f, k)\mathbf{G}(k, \tau) \quad (5)$$

where $\mathbf{Y}(f, \tau)$ is the magnitude spectrogram of the synthetic signal, $\hat{\mathbf{Y}}(f, \tau)$ is the estimated spectrogram, $\mathbf{G}(k, \tau)$ matrix represents the gain of the spectral pattern

¹ MIDI synthetic signals are generated using Timidity++ with the FluidR3 GM soundfont on Mac OS

for combination k at frame τ , and $\mathbf{B}(f, k)$ matrix, for $k = 1, \dots, K$, represents the spectral patterns for all the combinations of notes defined in the score.

When the parameters are restricted to be non-negative, as it is the case of magnitude spectra, a common way to compute the factorization is to minimize the reconstruction error between the observed spectrogram and the modeled one.

The most popular cost functions are the Euclidean (EUC) distance, the generalized Kullback-Leibler (KL) and the Itakura-Saito (IS) divergences.

Besides, the Beta-divergence (see eq. 6) is another commonly used cost function that includes in its definition the three previously mentioned EUC ($\beta = 2$), KL ($\beta = 1$) and IS ($\beta = 0$) cost functions.

$$D_\beta(x\hat{x}) = \begin{cases} x \log \frac{x}{\hat{x}} - x + \hat{x} & \beta = 1 \\ \frac{x}{\hat{x}} + \log \frac{x}{\hat{x}} - 1 & \beta = 0 \\ \frac{1}{\beta(\beta-1)} (x^\beta + (\beta-1)\hat{x}^\beta - \beta x\hat{x}^{\beta-1}) & \text{otherwise,} \end{cases} \quad (6)$$

In order to obtain the model parameters that minimize the cost function, Lee *et al.* [18] proposes an iterative algorithm based on MU rules. Under these rules, $D_\beta(\mathbf{Y}(f, \tau) | \hat{\mathbf{Y}}(f, \tau))$ is shown to be non-increasing at each iteration while ensuring non-negativity of the bases and the gains. Details are omitted to keep the presentation compact, for further information please read [18, 19]. For the model of eq. (5), multiplicative updates which minimize the Beta-divergence are defined as

$$\mathbf{B}(f, k) \leftarrow \mathbf{B}(f, k) \odot \frac{\left(\mathbf{Y}(f, \tau) \odot \hat{\mathbf{Y}}^{\beta-2}(f, \tau)\right) \mathbf{G}^T(\tau, k)}{\left(\hat{\mathbf{Y}}^{\beta-1}(f, \tau)\right) \mathbf{G}^T(\tau, k)} \quad (7)$$

$$\mathbf{G}(k, \tau) \leftarrow \mathbf{G}(k, \tau) \odot \frac{\mathbf{B}(f, k) \left(\mathbf{Y}(f, \tau) \odot \hat{\mathbf{Y}}^{\beta-2}(f, \tau)\right)}{\mathbf{B}(f, k) \left(\hat{\mathbf{Y}}^{\beta-1}(f, \tau)\right)} \quad (8)$$

where operator \odot indicates Hadamard product (or element-wise multiplication), division and power are also element-wise operators and $(\cdot)^T$ denotes matrix transposition.

Finally, the method to learn the spectral patterns for each state is described in Algorithm 1.

Algorithm 1 Method for learning spectral patterns combinations

- 1 Initialize $\mathbf{G}(k, \tau)$ as the combinations activation matrix $\mathbf{R}(k, \tau)$ and $\mathbf{B}(f, k)$ with random positive values.
 - 2 Update the bases using eq. (7).
 - 3 Update the gains using eq. (8).
 - 4 Normalize each spectral pattern of $\mathbf{B}(f, k)$ to the unit β -norm.
 - 5 Repeat step 2 until the algorithm converges (or maximum number of iterations is reached).
-

As explained in Section 3.1.1, $\mathbf{R}(k, \tau)$ is a binary combination/time matrix that represents the activation of combination k at frame τ of the training data. Therefore, at each frame, the active combination k is set to one and the rest are zero. Gains initialized to zero will remain zero, and therefore the frame becomes represented with the correct combination.

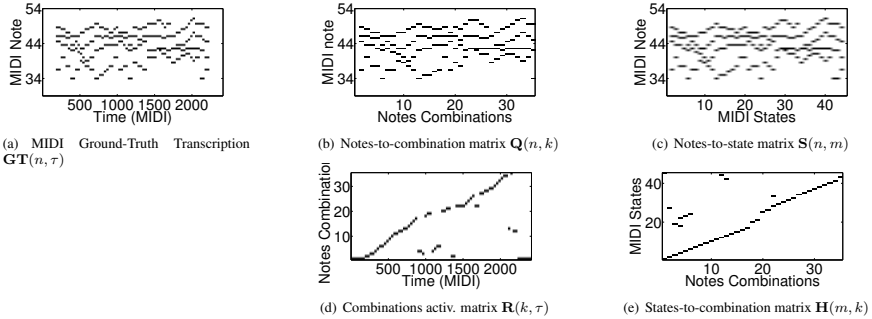


Figure 2: Music signal from the test database in Section 4 (“01-AchGottundHerr”). (a) MIDI Ground-Truth Transcription $\mathbf{GT}(n, \tau)$. (b) Notes-to-combination matrix $\mathbf{Q}(n, k)$. (c) Notes-to-state matrix $\mathbf{S}(n, m)$. (d) Combinations activation matrix $\mathbf{R}(k, \tau)$. (e) States-to-combination matrix $\mathbf{H}(m, k)$.

3.2 Alignment Stage

In this stage, the alignment between the score and the audio performance is accomplished in realtime using the information from the preprocessing stage.

3.2.1 Observation Model

As explained in Section 3.1.2, the spectral patterns $\mathbf{B}(f, k)$ for the K different combinations of notes are learned in advance using a MIDI synthesizer and kept fixed. Each spectral pattern models the spectrum of a unique combination.

Now, the aim is to compute the gain matrix $\mathbf{G}(k, t)$ and the cost matrix $\mathbf{D}(\tau, t)$ that measures the suitability of each combination of notes belonging to each MIDI time τ to be active at each frame t (referenced to the signal input) by analyzing the similarity between the spectral patterns $\mathbf{B}(f, k)$ and the input signal spectrogram². From the cost matrix $\mathbf{D}(\tau, t)$, a classical DTW approach can be applied to compute the alignment path.

In this work, we propose to perform the factorization using the realtime single-pitch constrained method proposed in [15]. Although this method was designed to address music transcription of monophonic signals, it can be adapted for audio to score alignment of polyphonic signals because only one combination will be active at a time. In this transcription method, the optimum combination k_{opt} is chosen to minimize the Beta-divergence function at frame t under the assumption that only one gain is non-zero at each frame. Taking the combinations as the index of gains $\mathbf{G}(k, t)$, this assumption is fair because only a unique combination k of notes is active at each time (at least when producing the audio signal).

Thus, the signal model with the single-combination constraint for the signal input vector at time t , $\mathbf{x}_t(f)$, is defined as follows.

$$\mathbf{x}_t(f) \approx \tilde{\mathbf{x}}_{k_{opt}, t}(f) = g_{k_{opt}, t} \mathbf{b}_{k_{opt}}(f) \quad (9)$$

²Note that we are using \mathbf{X} and t instead of \mathbf{Y} and τ to represent the signal magnitude spectrogram and the time frames to distinguish between real world and synthetic signals.

where $\tilde{\mathbf{x}}_{k_{opt}, t}(f)$ is the modeled signal for the optimum combination k_{opt} at frame t .

$$k_{opt}(t) = \arg \min_{k=1, \dots, K} D_{\beta}(\mathbf{x}_t(f) | g_{k, t} \mathbf{b}_k(f)) \quad (10)$$

The signal model in eq. (9) assumes that when combination k is active all other combinations are inactive and, therefore, the gain $g_{k, t}$ is just a scalar and represents the gain of the k combination. The model of eq. (10) allows the gains to be directly computed from the input data $\mathbf{X}(f, t)$ and the trained spectral patterns $\mathbf{B}(f, k)$ without the need of an iterative algorithm and thus, reducing the computational requirements. To obtain the optimum combination at each frame, we must first compute the divergence obtained by the projection of each combination at each frame and then select the combination that achieves the minimum divergence as the optimum combination at each frame.

In the case of Beta-divergence, the cost function for combination k and frame t can be formulated as

$$D_{\beta}(\mathbf{x}_t(f) | g_{k, t} \mathbf{b}_k(f)) = \sum_f \frac{1}{\beta(\beta-1)} (\mathbf{x}_t^{\beta}(f) + (\beta-1)(g_{k, t} \mathbf{b}_k(f))^{\beta} - \beta \mathbf{x}_t(f)(g_{k, t} \mathbf{b}_k(f))^{\beta-1}) \quad (11)$$

The value of the gain for combination k and frame t is then computed by minimizing eq. (11). Conveniently, this minimization has a unique non-zero solution due to the scalar nature of the gain for combination k and frame t (see more details in [15]).

$$g_{k, t} = \frac{\sum_f \mathbf{x}_t(f) \mathbf{b}_k(f)^{(\beta-1)}}{\sum_f \mathbf{b}_k(f)^{\beta}} \quad (12)$$

Finally, the divergence matrix for each combination at each frame is defined as:

$$\Phi(k, t) = D_{\beta}(\mathbf{x}_t(f) | g_{k, t} \mathbf{b}_k(f)) \quad (13)$$

where β can take values in the range $\in [0, 2]$.

As can be inferred, the divergence matrix $\Phi(k, t)$ provides us information about the similitude of each combination k spectral pattern with the real signal spectrum at each frame t . Using this information, we can directly compute the cost matrix between the MIDI time τ and the time of the input signal t as

$$\mathbf{D}(\tau, t) = \mathbf{R}^T(\tau, k)\Phi(k, t) \quad (14)$$

where $\mathbf{R}(k, \tau)$ is the combinations activation matrix defined in Section 3.1.1 and superscript “T” stands for matrix transposition. The process is detailed in Algorithm 2.

Algorithm 2 Divergence matrix computation method

```

1 Initialize  $\mathbf{B}(f, k)$  with the values learned in Section 3.1.2.
2 for  $t=1$  to  $T$  do
3   for  $k=1$  to  $K$  do
4     Compute the gains  $g_{k,t}$  using eq. (12).
5     Compute the current value the divergence matrix
        $\Phi(k, t)$  using eq. (13).
6   end for
7 end for
8 Compute the cost matrix  $\mathbf{D}(\tau, t)$  between MIDI time and input
  signal time using (14).
```

To resume, we propose the use the divergence matrix $\mathbf{D}(\tau, t)$ as the input of the DTW algorithm in order to perform the alignment.

3.2.2 Path Computation

We here propose to use a DTW based method to perform the alignment using the cost matrix $\mathbf{D}(\tau, t)$ obtained in eq. (14). This cost matrix is computed from the input signal $\mathbf{X}(f, t)$ and the “synthetic” spectral patterns per combination $\mathbf{B}(f, k)$ explained in Section 3.1.2. The term “synthetic” comes from the fact that the spectral patterns $\mathbf{B}(f, k)$ are computed from the score using a MIDI synthesizer.

a) Offline approach: This approach represents the classical offline alignment using DTW. To this end, we have used the code from [23]. The forward step is computed as in the classical DTW (see eq. (2)). In this experiment, c_i and c_j range from 1 to 4 in order to allow 4 times faster speed of interpretation. Finally the optimum path is obtained by tracing the recursion backwards from $\mathbf{C}(I, J)$ as in the original formulation of DTW (see Section 2).

b) Online approach: The online algorithm differs from an standard (i.e. offline) DTW algorithm in some points. Firstly, the signal is partially unknown (or the future of the signal is not known when making the alignment decisions), so the global path constraints cannot be directly implemented, in other words, the recursion backwards can not be traced from the last frame T of the signal. Secondly, if some latency (i.e. delay in the decision) is permitted, the recursion backwards can be traced in equally spaced frames of the input signal making the latency equal to the difference in time of the frame when the backtracking is done and the input signal frame. Finally, in order to run

in realtime, the complete algorithm should not increase the complexity with the length of the signal.

In this work, we used the online scheme proposed by Dixon in [13]. In fact, Dixon’s algorithm calculates an “adaptive diagonal” through the cost matrix by seeking the best path considering a searching band with a fixed width. Here, we propose an online algorithm with a fixed latency of just one frame. In order to obtain this low latency, no backtracking is allowed, taking the decision directly from the forward information at each frame t . As a consequence of the low latency of online algorithms (apart from the complexity reduction), the obtained results are degraded from their offline counterparts. In fact, for those situations in which a higher latency can be supported, delaying the decision in time using a limited traceback can improve the obtained results of the online algorithms.

4. EXPERIMENTAL SETUP

a) Time-Frequency representation: In this paper we use a low-level spectral representation of the audio data which is generated from a windowed FFT of the signal. A Hanning window with the size of 128 ms, and a hop size of 10 ms is used (for both synthetic and real-world signals). Here, we use the resolution of a single semitone as in [21]. In particular, we implement the time-frequency representation by integrating the STFT bins corresponding to the same semitone.

b) Evaluation metrics: We have used the same evaluation metrics as in the MIREX Score Following task. Detailed information can be found in [22]. For each piece, aligned rate (AR) or precision is defined as the proportion of correctly aligned notes in the score and ranges from 0 to 1. A note is said to be correctly aligned if its onset does not deviate more than a threshold (a.k.a tolerance window) from the reference alignment. Missed notes are events that are present in the reference but not reported. Recognized notes whose onsets are far from the given threshold are considered misaligned notes.

c) Dataset: The dataset used to evaluate our method is comprised of excerpts from 10 human played J.S. Bach four-part chorales. The audio files are sampled from real music performances recorded at 44.1 kHz that are 30 seconds in length per file. Each piece is performed by a quartet of instruments: violin, clarinet, tenor saxophone and bassoon. Each musician’s part was recorded in isolation. Individual lines were then mixed to create 10 performances with four-part polyphony. Ground-truth alignment is provided for both, individual sources and mixture, the latter assuming constant tempo between annotated beats and a perfect synchronization between the musicians. More information about this dataset can be found in [6].

5. RESULTS

To analyze the performance of the proposed (offline and online) methods in Section 3. Evaluation has been per-

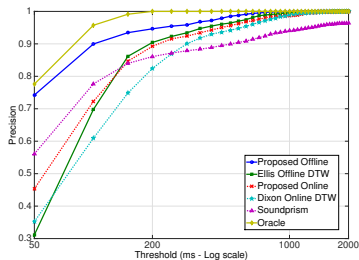


Figure 3: Precision values in function of the tolerance window

formed using the metrics detailed in Section 4. The proposed systems are compared with four reference methods that are detailed below: a) Ellis’ Offline DTW [23] [24], b) Dixon’s Online DTW [13], c) Soundprism [6] and d) Oracle. Note that the latter is not a score following system but provided aligned MIDI score assuming constant tempo between annotated beats and perfect synchronization between musicians as explained in Section 4. The evaluation of this oracle information is very interesting to analyze the deviation of the different instrument performances between themselves and also to measure the best performance that can be obtained by score followers that are only capable of aligning on a global level (i.e., cannot detect the onset positions of individual notes). Note that there are several works that attempt to refine the alignment by synchronizing the onsets/offsets individually for each note in a post-processing stage [25–27].

In a first experiment, we have evaluated the precision of the analyzed methods as a function of the onset deviation threshold. To this end, the threshold value was varied from 50 to 2000 ms in 50 ms steps. Obtained results are plotted in Figure 3. As can be seen, the Oracle method, which can be considered as the upper bound for score followers performing global alignment, requires around 200 ms threshold to obtain a perfect alignment. This value comes from the difference between the ground-truth alignment for each instrument played in isolation and the global ground-truth of the whole mixture, obtained by interpolating the annotated beat times of each audio.

In general, our offline approach obtain the best results in terms of precision. In fact, our offline approach clearly outperforms Ellis’ offline approach, mainly due to the factorization based feature extraction stage. Regarding the online methods, our online approach and Soundprism obtain similar results on average than the Ellis’ offline approach and clearly outperform Dixon’s online approach. Soundprism seems to perform better when using lower threshold values while our online approach allows convergence to the optimum alignment as the threshold is increased.

In a second experiment (see Table 1), we evaluate the performance of the proposed methods as a function of the polyphony. A fixed threshold (a.k.a tolerance window) of 200ms is used because, as illustrated in Figure 3, this value represents the difference between isolated instruments and

	Poly	Precision	Miss	Missalign	Av offset	Av offset	Std Offset
Prop. Offline	2	94.59	0.00	5.41	-11.27	33.43	44.38
	3	94.75	0.00	5.25	-11.78	34.25	44.89
	4	94.50	0.00	5.50	-11.09	35.16	46.51
Ellis Offline	2	90.57	0.00	9.42	66.90	71.28	47.30
	3	90.39	0.00	9.60	65.67	71.53	50.24
	4	89.80	0.00	10.19	66.97	72.62	49.93
Prop. Online	2	88.44	0.00	11.56	-41.18	57.78	56.40
	3	90.13	0.00	9.86	-42.96	60.56	57.65
	4	90.70	0.00	9.30	-44.15	62.97	58.58
Dixon Online	2	81.69	0.00	18.31	53.93	70.51	67.02
	3	83.17	0.00	16.83	53.88	70.65	67.02
	4	83.94	0.00	16.06	51.29	71.64	70.26
Soundprism	2	83.01	0.00	16.99	-25.90	48.08	55.36
	3	88.81	0.00	11.19	-21.23	43.73	51.54
	4	93.50	0.00	6.50	-22.05	42.91	49.13
Oracle	2	100.00	0.00	0.00	6.15	32.79	42.47
	3	100.00	0.00	0.00	6.09	32.67	43.08
	4	100.00	0.00	0.00	6.07	32.58	43.38

Table 1: Audio-to-score results as a function of polyphony in terms of piecewise precision (%). Offset values in ms. The bold percentage shows the best result for each measure.

mixture ground-truth alignment.

As explained in the previous section, offline methods perform better in general than the online ones. The proposed offline method obtains the best results among the compared methods and has demonstrated to be robust against polyphony in the analyzed dataset (polyphony 2 to 4). Regarding the online methods, our online approach and Soundprism obtain similar results on average and clearly outperforms Dixon’s online approach, although the former seems to be more robust against the polyphony.

In relation to the offset, the oracle solution exhibits the minimum possible std offset due to the differences in starting times for the same states between musicians. Moreover, our offline approach and the online Soundprism have the lower average offset values which means that both methods are more responsive and thus provide better results when dealing with lower thresholds.

6. CONCLUSIONS

In this paper we present a score following framework based on spectral factorization and DTW. Spectral factorization is used to learn spectral patterns for each combination of concurrent notes in the MIDI score. Then, a cost matrix is computed using the divergence matrix obtained using a non-iterative signal decomposition method previously developed by the authors in [15] that has been tuned to perform the projection of each combination of notes. Finally, a DTW strategy is performed in an offline and online manner. The proposed offline and online approaches have been tested using a dataset with different polyphony levels (from 2 to 4) and compared them with other reference methods. On average, our approaches (offline and online) obtain the best results in terms of precision within the compared offline and online approaches, respectively, and has demonstrated to be robust against the analyzed polyphony.

In the future we plan to track the tempo changes in order to enforce a certain degree of continuity in the online decisions. Besides, we will extend the evaluation of our method using a larger dataset of a varied range of instruments, dynamics and different styles.

7. REFERENCES

- [1] A. Cont, "A coupled duration-focused architecture for real-time music-to-score alignment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 6, pp. 974987, Jun. 2010.
- [2] N. Montecchio and N. Orio, "A discrete filterbank approach to audio to score matching for score following," in *Proc. ISMIR*, 2009, pp. 495-500.
- [3] N. Hu, R. B. Dannenberg, and G. Tzanetakis, "Polyphonic audio matching and alignment for music retrieval," in *Proc. IEEE WASPAA*, 2003, pp. 185-188.
- [4] O. Izmirli and R. Dannenberg, "Understanding features and distance functions for music sequence alignment". In *Proceedings of ISMIR*, 411- 416. 2010
- [5] M. Puckette, "Score following using the sung voice," in *Proc. ICMC*, 1995, pp. 175-178.
- [6] Z. Duan and B. Pardo, "Soundprism: An Online System for Score-informed Source Separation of Music Audio," *IEEE Journal of Selected Topics in Signal Process.*, vol. 5, no. 6, pp. 1205-1215, 2011.
- [7] P. Cano, A. Loscos, and J. Bonada, "Score-performance matching using HMMs," in *Proc. ICMC*, 1999, pp. 441-444.
- [8] A. Cont, "Realtime audio to score alignment for polyphonic music instruments using sparse non-negative constraints and hierarchical hmms." In *Proceedings of IEEE ICASSP*, Toulouse, France, 2006.
- [9] C. Raphael, "Automatic segmentation of acoustic musical signals using hidden Markov models," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 4, pp. 360370, Apr. 1999.
- [10] C. Joder, S. Essid, and G. Richard. "Learning optimal features for polyphonic audio-to-score alignment." *IEEE Transactions on Audio, Speech, and Language Processing*, 21, 10, 2118-2128, 2013
- [11] P. Cuvillier and A. Cont. "Coherent time modeling of Semi-Markov models with application to realtime audio-to-score alignment". *Proceedings of the 2014 IEEE International Workshop on Machine Learning for Signal Processing*, 16, 2014.
- [12] N. Orio and D. Schwarz, "Alignment of monophonic and polyphonic music to a score," in *Proc. International Computer Music Conference (ICMC)*, 2001.
- [13] S. Dixon, "Live tracking of musical performances using on-line time warping," in *Proc. International Conference on Digital Audio Effects (DAFx)*, Madrid, Spain, 2005, pp. 92-97.
- [14] J. Fritsch and M. Plumbley, "Score Informed Audio Source Separation using Constrained Nonnegative Matrix Factorization and Score Synthesis," In *Proc. ICASSP*, Vancouver, Canada, 2013.
- [15] J.J. Carabias-Orti et al., "Constrained non-negative sparse coding using learnt instrument templates for real-time music transcription," *Engineering Applications of Artificial Intelligence*, April 2013
- [16] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 23, pp. 5272, 1975.
- [17] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimisation for spoken word recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 26, pp. 4349, 1978.
- [18] D. D. Lee and H. S. Seung, "Algorithms for Non-negative Matrix Factorization," in *Proc. of Neural Information Processing Systems*, Denver, USA, 2000.
- [19] C. Févotte and J. Idier "Algorithms for Nonnegative Matrix Factorization with the Beta-Divergence," *Neural Computation*, vol. 23, no. 9, pp. 2421-2456, September 2011.
- [20] J. F. Gemmeke et al., "Exemplar-based sparse representations for noise robust automatic speech recognition," *IEEE Trans. Audio, Speech and Language Processing*, Volume: 19, Issue: 7, 2011.
- [21] J.J. Carabias-Orti et al., "Musical Instrument Sound Multi-Excitation Model for Non-Negative Spectrogram Factorization," *IEEE Journal of Selected Topics in Signal Processing*, Vol. 5, no. 6, pp. 1144 - 1158, October 2011
- [22] A. Cont et al., "Evaluation of real-time audio-to-score alignment," in *Proc. International Conference on Music Information Retrieval (ISMIR)*, 2007.
- [23] D. Ellis. Dynamic Time Warp (DTW) in Matlab, Web resource, available: <http://www.ee.columbia.edu/~dpwe/resources/matlab/dtw/>.
- [24] R. Turetsky and D. Ellis "Ground-Truth Transcriptions of Real Music from Force-Aligned MIDI Syntheses", in *Proc. International Conference on Music Information Retrieval (ISMIR)*, 2003.
- [25] B. Niedermayer, and G. Widmer. "A multi-pass algorithm for accurate audio-to-score alignment" in *Proc. International Conference on Music Information Retrieval (ISMIR)*, 2010.
- [26] J. Devaney, "Estimating Onset and Offset Asynchronies in Polyphonic Score-Audio Alignment" *Journal of New Music Research* 43 (3), 266-275, 2014
- [27] M. Miron, J. Carabias-Orti and J. Janer. "Improving Score-Informed Source Separation Through Audio To Score Note-Level Refinement " in *Proc. International Conference on Music Information Retrieval (ISMIR)*, 2015.

NEW SONORITIES FOR EARLY JAZZ RECORDINGS USING SOUND SOURCE SEPARATION AND AUTOMATIC MIXING TOOLS

Daniel Matz

University of Applied Sciences
Düsseldorf, Germany
daniel.matz@hotmail.de

Estefanía Cano

Fraunhofer IDMT
Ilmenau, Germany
cano@idmt.fhg.de

Jakob Abeßer

Fraunhofer IDMT
Ilmenau, Germany
abr@idmt.fhg.de

ABSTRACT

In this paper, a framework for automatic mixing of early jazz recordings is presented. In particular, we propose the use of sound source separation techniques as a pre-processing step of the mixing process. In addition to an initial solo and accompaniment separation step, the proposed mixing framework is composed of six processing blocks: harmonic-percussive separation (HPS), cross-adaptive multi-track scaling (CAMTS), cross-adaptive equalizer (CAEQ), cross-adaptive dynamic spectral panning (CADSP), automatic excitation (AE), and time-frequency selective panning (TFSP). The effects of the different processing steps in the final quality of the mix are evaluated through a listening test procedure. The results show that the desired quality improvements in terms of sound balance, transparency, stereo impression, timbre, and overall impression can be achieved with the proposed framework.

1. INTRODUCTION

When early jazz recordings are analyzed from a modern audio engineering perspective, clear stylistic differences can be identified with respect to modern recording techniques. These characteristics mainly evidence the technological and stylistic differences between the two eras. For example, solo instruments such as the saxophone or the trumpet often completely dominate the audio mix in early jazz recordings. At the same time, the rhythm section, i.e., double bass, piano, drums, and percussion, often falls in a secondary place, recorded or mixed with much lower intensity and often perceived as unclear and undifferentiated. Additionally, from today's perspective, early jazz recordings often present an unusual stereo image. Instrument groups are sometimes assigned to extreme stereo positions which can cause the solo instrument to be panned to the left and the accompaniment band panned to the right. As a consequence, the energy distribution over the stereo width is unbalanced and is often perceived today as irritating and disturbing, especially when listened through headphones.



© Daniel Matz, Estefanía Cano, Jakob Abeßer.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Daniel Matz, Estefanía Cano, Jakob Abeßer. "New sonorities for early jazz recordings using sound source separation and automatic mixing tools", 16th International Society for Music Information Retrieval Conference, 2015.

Several initiatives have arisen that attempt to give such early recordings a more modern sonority. Remastering and Automatic Mixing (AM) techniques offer various methods for a sonic redesign of such recordings. However, given that the original individual stems of the instruments in the recordings are usually not available, these techniques can only achieve minor modifications to the sound characteristics of mono and stereo mixtures. In-depth remixing usually requires the original multi-track recordings to be available. For this purpose, sound source separation methods developed in the Music Information Retrieval (MIR) community can be useful tools to retrieve individual instruments from a given mix.

2. GOALS

The main goal of this study is to identify suitable signal processing methods to modify the above-mentioned characteristics in a selection of early jazz recordings. These methods are combined in a fully automatic mixing framework. In particular, we focus on modifying the audio mix in terms of transparency, stereo impression, frequency response, and acoustic balance in order to improve the overall perception of sound and the quality of the mix with respect to the original recording.

Our main approach for remixing is to modify the characteristic of the backing track to make it more present in the mix. We also aim at improving the acoustical and spatial balance of the audio mix. The solo signal is balanced with respect to its loudness and spectral energy to minimize spectral masking as well as to improve its position in the stereo image.

3. RELATED WORK

In the field of automatic mixing, several approaches have been presented in the literature. In [1], a method is proposed to automatically adjust gain and equalizer parameters for multi-track recordings using a least-squares optimization. In [12] the idea of modifying the magnitude spectrogram of a signal towards a target spectrogram called *target mixing*, is presented. Other approaches for automatic mixing of multi-track recordings have incorporated machine learning algorithms to perform the mixing process [16, 17].

In [14] and [19], several cross-adaptive signal processing methods for automatic mixing such as source enhancer, panner, fader, equalizer, and polarity and time offset correction are proposed. These modules can be combined into a full mixing application. In [4], the authors propose a knowledge-engineered autonomous mixing system and propose to include expert knowledge within an automatic mixing system. The included audio effects are automatically controlled based on extracted low-level and high-level features such as musical genre, instrumentation, and the type of sound sources. The authors evaluated the system using short four bar audio signals with vocals, bass, guitar, keyboard, and other instruments.

Harmonic-percussive source separation was used as pre-processing step for manual remixing in [6], in particular to adjust the sound source levels of the signals. To the authors' best knowledge, a framework for automatic remixing that suits the requirements discussed in section 2 has not been proposed so far.

4. PROPOSED METHOD

For our mixing framework, we propose the use of sound source separation techniques as a pre-processing step of the mixing process. For this purpose, we first isolate the solo instrument from the audio mix by applying an algorithm for pitch-informed solo and accompaniment separation [2]. The two separated signals, i.e., the *solo* and the *residual/backing* signal, are the starting point for the automatic remixing process. Additionally, based on the requirements discussed in section 2, our proposed framework comprises six subcomponents:

1. Harmonic-percussive separation (HPS)
2. Cross-adaptive multi-track scaling (CAMTS)
3. Cross-adaptive equalization (CAEQ)
4. Cross-adaptive dynamic spectral panning (CADSP)
5. Automatic excitation (AE)
6. Time-frequency selective panning (TFSP)

Figure 1 presents a block diagram of the proposed framework. There are three main signal pathways A, B, and C. If the CADSP is activated, pathway A is chosen. If CADSP is not activated, pathway B and C are chosen depending on whether the harmonic-percussive separation (HPS) is used. All signal paths output a stereo mix. In the following sections, the individual subcomponents are first described, followed by a description of the three proposed signal pathways.

4.1 Solo and Backing track Separation

The algorithm as proposed in [2] automatically extracts pitch sequences of the solo instrument and uses them as prior information in the separation scheme. In order to

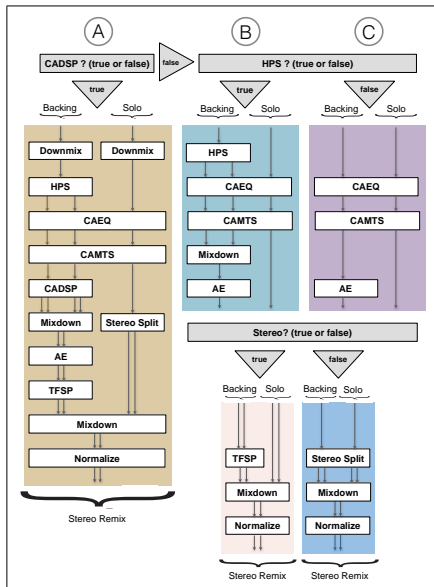


Figure 1: Signal flow-chart of the developed automatic remixing framework

obtain more accurate spectral estimates of the solo instrument, the algorithm creates tone objects from the pitch sequences, and performs separation on a tone-by-tone basis. Tone segmentation allows more accurate modeling of the temporal evolution of the spectral parameters of the solo instrument. The algorithm performs an iterative search in the magnitude spectrogram in order to find the exact frequency locations of the different partials of the tone. A smoothness constraint is enforced on the temporal envelopes of each partial. In order to reduce interference from other sources caused by overlapping of spectral components in the time-frequency representation, a common amplitude modulation is required for the temporal envelopes of the partials. Additionally, a post-processing stage based on median filtering is used to reduce the interference from percussive instruments in the solo estimation.

4.2 Harmonic-percussive Separation (HPS)

We use the algorithm for harmonic-percussive separation proposed in [6]. The algorithm is based on median filtering of the magnitude spectrogram to split the original audio signal into its horizontal (harmonic sources) and vertical elements (percussive sources). In an automatic mixing context, these components can be understood as separate subgroups which can be processed individually and finally remixed.

4.3 Cross-adaptive Multi-track Scaling (CAMTS)

The method proposed in [19] which is commonly referred to as *automatic fader control*, is used for automatic scaling of the sound sources. The algorithm is used to automatically modify the amplification of separate sound sources. A psychoacoustic model based on the EBU R-128 standard [9] is used to compute the loudness of each track using a histogram-based approach. All tracks are individually amplified to be perceived as equally loud.

4.4 Cross-adaptive Equalizer (CAEQ)

We use the cross-adaptive equalizing algorithm proposed in [19] to obtain a spectrally balanced mixture. The main approach is to modify the spectral envelopes of the audio signals and to minimize the spectral masking between the solo signal and the backing track. The algorithm is a multi-band extensions of the CAMTS algorithm as discussed in section 4.3. The spectral characteristics of the separated signals are modified by enhancing or attenuating pre-defined frequency bands depending on the signal's perceived loudness with respect to the overall loudness. In contrast to the CAMTS algorithm, the loudness model proposed in [19] is used since it outperformed the loudness model based on EBU R-128 during informal testing. In particular, the mix results based on EBU-R 128 showed too strong of an emphasis on treble frequencies while lacking energy in the lower frequency range. We use a 10-band octave equalizer with second-order biquad IIR filters following [19] and frequency bands uniformly distributed over the audible frequency range. Standard frequency values based on [8] are used to adjust the center frequencies of the peak filter as well as the cutoff frequencies of the shelving filters.

4.5 Cross-adaptive Dynamic Spectral Panning (CADSP)

Dynamic spectral panning is a technique that allows the creation of a stereophonic impression in a given monophonic multi-track recording. We use the algorithm proposed in [15] to create a spatialization effect given multi-track signals. The method dynamically assigns time-frequency bins of the original tracks towards azimuth positions. The assignment reduces masking due to shared azimuth positions between multiple sound sources. This improves the overall transparency of an audio mix. In the cases where the original audio mix is a stereo track, it is first down-mixed to mono and then up-mixed to a new stereo image using the CADSP algorithm.

4.6 Automatic Exciter (AE)

The exciting algorithm improves the assertiveness of the backing track. The digital signal processing methods are implemented following the *APHEX Aural Exciter* described in [18]. The audibility of the mixed signal is enhanced by adding harmonic distortions in the upper frequency range. These distortions create additional harmonic signal com-

ponents which improve the presence, clarity, and brightness of the audio signal.

The automation of the exciting step is implemented following a *target mixing* approach. Based on [5], the mixing parameters are iteratively adjusted to a *target energy ratio*. The target energy ratio is computed from the relationship between the energy of the high-pass filtered signal and the energy of the target signal. In the *side chain*, an asymmetric soft clipping characteristic, *harmonic generator block*, with adaptive threshold was used. This allows a level-independent distortion as well as the preservation of the signal dynamics [5].

4.7 Time-frequency selective Panning (TFSP)

Time-frequency selective panning improves the stereo image as well as the overall spatial impression of an audio mix. In our framework, the method for time-frequency selective panning presented in [3] was used. The azimuth positions of the sound sources are modified using a non-linear *warping function*. The stereo image is widened while the initial arrangement of the sound sources, as well as the sound quality of the original source is maintained. Within the proposed automatic remixing framework, the TFSP algorithm can be interpreted as an extension of the CADPS algorithm. The panning algorithm is only applied to the residual signal (see section 4.8.1). We set the aperture parameter ρ to a fixed value based on initial informal testing.

4.8 Processing Pathways

4.8.1 Signal path A (Main Path)

The main processing path includes all system components. Stereo files must be down-mixed to mono first due to constraints of the *cross-adaptive dynamic spectral panning* (CADSP) algorithm as detailed in section 4.5. All sound sources, which are initially distributed in the stereo panorama, are first centered to the mono channel and later re-distributed over the stereo panorama again based on the harmonic-percussive sound separation [6]. This up-mixing step that can involve a modification of the stereo arrangement is only possible in this signal path.

The *cross-adaptive equalization* (CAEQ) and *multi-track scaling* (CAMTS) are the first processing steps in all three pathways. After applying the *dynamic spectral panning* (CADSP) to the percussive and harmonic signal components, all stereophonic signals are summed up to a backing track with a more homogeneous distribution of the sound sources. The backing track can now be processed with the *automatic excitation* (AE) and the *time-frequency selective panning* (TFSP) algorithms. The solo signal is split into stereo channels in the *Stereo Split* stage and scaled such that the overall gain remains constant. In the final *mix-down* step, the backing track is mixed with the solo track by adjusting the individual amplification factors as given by the CAMTS stage. If the cross-adaptive equalization (CAEQ) was performed, the spectral envelope of the backing track is perceptibly modified due to the minimization

of the spectral masking. The stereo sum signal is finally *normalized*.

4.8.2 Signal path B

Signal path B resembles signal path A, however, the equalization (CAEQ) and scaling (CAMTS) steps offer more ways to modify parameters due to the prior harmonic-percussive separation stage.

4.8.3 Signal path C

In the signal path C, no harmonic-percussive separation is performed. The equalization (CAEQ) and scaling (CAMTS) are applied to both the backing and the solo track. However, the automatic excitation is only applied to the backing track since we particularly want to enhance the presence, clarity, and brightness of the backing track. As shown in figure 1, the time-frequency selective panning (TFSP) can only be applied to the backing track if it is a stereo signal. For monaural signals, the signal is split to the stereo channels (*Stereo Split*) and scaled such that the overall gain remains constant. Similar to signal path B, the signals are finally mixed down and normalized.

5. EVALUATION

5.1 Experimental Design

To evaluate the proposed framework, a listening test procedure was conducted following the guidelines of the *Multi Stimulus Test with Hidden Reference and Anchor* (MUSHRA) described in the ITU-R BS.1534-2 recommendation [11], and modifying them to fit the characteristics of this study. The main difference of our test with respect to the original MUSHRA is that a reference signal, which in our case would be an ideal mix of the original recording, is not available. Moreover, the notion of an ideal mix is ill-posed in the automatic remixing context.

The listening test was conducted in a quiet room and all signals were played using open headphones (AKG K 701). A total of 19 participants conducted the listening test. The participants included audio signal processing experts, professional audio engineers, music students (jazz, classical music), musicologists, as well as amateur musicians and regular music consumers. The average age of the participants was 30.7 years old. Further demographic information such as gender, hearing impairments, listening test experience, and educational background were also collected. A summary of the demographic information is presented in table 1.

The listening test was divided into five evaluation tasks, each focusing on a different subjective quality parameter. The following parameters were selected based on the ITU-R BS.1248-1 recommendation [10], and were adopted to our requirements: (QP1) Sound Balance, (QP2) Transparency, (QP3) Stereo/Spatial Impression, (QP4) Timbre, and (QP5) Overall Impression. In each evaluation task, a *training phase* was first conducted to allow the participants to familiarize themselves with the test material and to adjust playback levels to a comfortable one.

Gender	M	16
	F	3
Hearing impairment?	Yes	0
	No	19
Listening test experience?	Yes	9
	No	10
Expert in audio engineering?	Yes	11
	No	8
Educational background in music?	Yes	15
	No	4

Table 1: Demographics of the listening test participants

Following the training phase, an *evaluation phase* was conducted for each task. Five audio tracks as described in Table 2 with ten mixtures each were rated by the participants. The five tracks used in this study are part of the Jazzomat Database¹. Among the presented mixtures, the original signal, eight mixes created with different configurations of the proposed framework, and an anchor signal (rhythm section reduced by 6 dB, the sum signal low-pass filtered at 3.5 kHz) were used. Table 3 gives an overview of all the remix configurations.

Title	Soloist (Instrument)	Style	Year
Body and Soul	Chu Berry (ts)	Swing	1938
Tenor Madness	Sonny Rollins (ts)	Hardbop	1956
Crazy Rhythm	J.J. Johnson (tb)	Bebop	1957
Bye Bye Blackbird	Ben Webster (ts)	Swing	1959
Adam's Apple	Wayne Shorter (ts)	Postbop	1966

Table 2: Dataset description

Mix	HPS	CAEQ	CAMTS	CADSP	AE	TFSP
1	off	on	off	off	on	off
2	off	off	on	off	on	off
3	off	on	on	off	on	off
4	on	on	off	off	on	off
5	on	off	on	off	on	off
6	on	off	off	on	on	on
7	on	on	on	on	on	on
8 (mono)	on	on	on	off	on	off

Table 3: Configurations of the eight remixes used in the listening test

The automatic exciting (AE) component is active in all the mixes. The panning (TFSP) algorithm is only activated in conjunction with the cross-adaptive dynamic spectral panning (CADSP). This way, a further stereo expansion of critical stereo recordings with an unbalanced stereo panorama is avoided. Mixture 8 was added to investigate the influence of the stereo effects (CADSP and TFSP) onto the input signals in the pre-processing step of pathway B that are mixed monophonic.

¹ A description of the Jazzomat Database is available at: <http://jazzomat.hfm-weimar.de/dbformat/dbcontent.html>

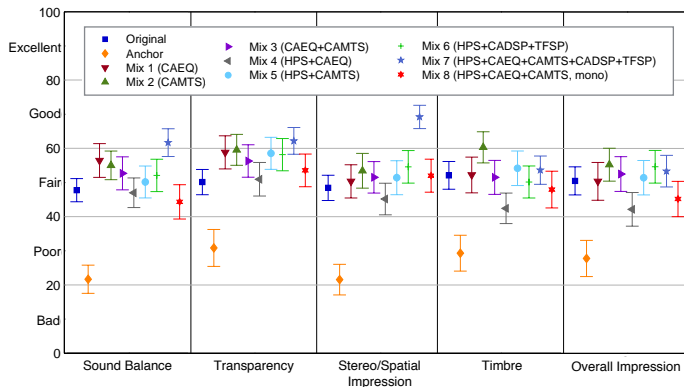


Figure 2: Listening test results for the five evaluated parameters.

5.2 Results

5.2.1 General

Figure 2 shows the results of the listening test for the five acoustic quality parameters. The figure legend summarizes all the system configurations that were evaluated. It is evident from the plot that the anchor stimulus was always correctly identified. Results also suggest that the use of harmonic-percussive separation does not bring perceptual quality gains for HPS+CAEQ (mix 4) when compared to the CAEQ (mix 3). Unexpectedly, results even got worse for the parameters timbre and overall impressions. Similarly, the combined settings in HPS+CAMTS (mix 5) do not show an improvement in the ratings when compared to CAMTS (mix 2).

To facilitate analysis of results, table 4 lists the percentage improvement obtained for each of the five quality parameters (QP), subject to the presence or absence of the individual framework components compared to the original signal. Mixes 4 and 5, which include the harmonic-percussive separation, were not listed due to the reasons previously described. The five mixtures listed in the table are further analyzed in the following sections.

	QP1	QP2	QP3	QP4	QP5
Mix 1 (CAEQ)	18%	17%	4%	-	-
Mix 2 (CAMTS)	15%	19%	10%	16%	9%
Mix 3 (CAEQ+CAMTS)	10%	12%	6%	-	4%
Mix 6 (HPS+CADSP+TFSP)	9%	16%	18%	-	8%
Mix 7 (All components)	29%	24%	43%	3%	6%

Table 4: Percentage improvement of the remixed signal compared to the original audio recording subject to the presence (or absence) of the individual framework components shown for each of the five perceptual quality parameters.

5.2.2 Mix 1 (CAEQ)

Mix 1 does not include a prior separation of the residual component and outperforms the original mix for most of the quality parameters. The highest improvements were 18% for sound balance and 17% for transparency. However, for timbre and overall impressions, no improvement was observed.

5.2.3 Mix 2 (CAMTS)

Despite the absence of the harmonic percussive separation step, mix 2 showed improvements for transparency (19%), sound balance (15%), and overall impression (9%). The reason for the improvement in timbre by 16% is not entirely clear in this case; however, a possible explanation is that the increased loudness of the rhythm section led to more balanced dynamic levels and a clearer perception of the instrument and overall timbres.

5.2.4 Mix 3 (CAEQ+CAMTS)

The combination of the CAEQ and CAMTS components showed inferior results compared to the exclusive application of both components. However, the ratings are still slightly higher than the ratings of the original audio file.

5.2.5 Mix 6 and Mix 7

Both mixtures 6 and 7 outperformed the original audio file. The highest ratings were achieved with mixture 7 which was extracted with the full processing chain. In particular, the improvements compared to the original audio file were 29% for sound balance, 24% for transparency, as well as 43% for stereo and spatial impression. The small improvements with respect to the overall impression are likely due to the individual aesthetic preferences of the listening test participants.

Additionally, to analyze the influence of the stereo effects to the input signals of pathway B (which are initially

downmixed to mono), Table 5 presents the percentage improvement obtained with mix 7 (all components active) in comparison to mix 8 (mono).

QP 1	QP 2	QP 3	QP 4	QP 5
39 %	16 %	33 %	12 %	18 %

Table 5: Mean ratings of the five quality parameters for the additional usage of the stereo effects (CADSP+TFSP) in mix 7 compared with the non-processed monophonic input signal in the same framework setting of mix 8 (HPS, CAEQ, CAMTS, AE).

As can be observed in the table, the use of the CADSP and TFSP modules improved the ratings for all five quality parameters. The improvement was statistically significant for sound balance (39 %) and stereo/spatial impression (33 %).

6. CONCLUSIONS

In this paper, we proposed a prototype implementation of an *automatic remixing framework* for tonal optimization of early jazz recordings. The main focus was on improving the balance between the solo instrument and the rhythm section. The framework consists of six components which include different processing steps to modify the loudness, frequency response, timbre, and stereophonic perception of the separated sound sources. We compared different configurations of the framework and evaluated the improvement of the transparency of the backing track as well as the acoustic balance, stereophonic homogeneity, and overall quality perception. The evaluation was performed with a MUSHRA-like listening test based on the ratings given by 19 participants.

The usage of automatic equalization (CAEQ) and multi-track scaling (CAMTS) showed clear improvement in the quality parameter ratings, whereas the combination of both led to a smaller improvements than the independent application of each approach. The improvement based on harmonic-percussive separation (HPS) within the automatic mixing framework is not easy to assess. The usage of HPS in conjunction with CAEQ and CAMTS did not improve the ratings. On the other hand, HPS is a basic requirement for the application of CADSP on the backing track of mix 7, and therefore contributes to its consistent high ratings. HPS is irrelevant for the automatic excitation (AE) step, since it is applied to the full residual track.

Particularly with mix 7 (all components), the initially targeted improvements in sound balance, stereo and spatial impression, and transparency with respect to the original audio recording were achieved.

In future work, the most relevant processing modules must be further investigated and improved with respect to the aforementioned quality parameters. Modules that showed none or only minor improvements must be replaced and alternative algorithms must be evaluated for the given tasks. Promising algorithms seem to be a mastering equalizer [7] or dynamic range compression [13]. The additional

use of semantic information of music genre and instrumentation seems to be another fruitful approach as discussed in section 3.

Finally, the integration of audio restoration methods such as denoising will likely help to remove unwanted background noise and spurious signals from the main signal to be processed.

7. REFERENCES

- [1] Daniele Barchiesi and Joshua D. Reiss. Automatic target mixing using least-square optimization of gains and equalization settings. In *Proceedings of the 12th International Conference on Digital Audio Effects (DAFx-09)*, Como, Italy, 2009.
- [2] Estefanía Cano, Gerald Schuller, and Christian Dittmar. Pitch-informed solo and accompaniment separation: towards its use in music education applications. *EURASIP Journal on Advances in Signal Processing*, 23:1–19, 2014.
- [3] Maximo Cobos and Jose J. Lopez. Interactive enhancement of stereo recordings using time-frequency selective panning. In *Proceedings of the 40th AES International Conference on Spatial Audio*, Tokyo, Japan, 2010.
- [4] Brecht De Man and Joshua D. Reiss. A semantic approach to autonomous mixing. *Journal of the Art of Record Production*, 8, 2013.
- [5] Brecht De Man and Joshua D. Reiss. Adaptive control of amplitude distortion effects. In *Proceedings of the 53rd AES International Conference on Semantic Audio*, London, UK, 2014.
- [6] Derry FitzGerald. Harmonic/percussive separation using median filtering. In *Proceedings of the 13th International Conference on Digital Audio Effects (DAFx)*, Graz, Austria, 2010.
- [7] Marcel Hilsamer and Stefan Herzog. A statistical approach to automated offline dynamic processing in the audio mastering process. In *Proceedings of the 17th International Conference on Digital Audio Effects (DAFx-14)*, Erlangen, Germany, 2014.
- [8] ISO International Organization for Standardization. Acoustics - preferred frequencies, August 1997.
- [9] ITU Radiocommunication Bureau. Algorithms to measure audio programme loudness and true-peak audio level (rec. itu-r bs.1770-3), August 2012.
- [10] ITU Radiocommunication Bureau. General methods for the subjective assessment of sound quality (rec. itu-r bs.1248-1), December 2003.
- [11] ITU Radiocommunication Bureau. Method for the subjective assessment of intermediate quality level of audio systems (rec. itu-r bs.1534-2), June 2014.

- [12] Zheng Ma, Joshua D. Reiss, and Black, Dawn A. A. Implementation of an intelligent equalization tool using yule-walker for music mixing and mastering. In *Proceedings of the 134th AES Convention*, Rome and Italy, 2013. AES.
- [13] Stylianos-Ioannis Mimilakis, Konstantinos Drossos, Andreas Floros, and Dionysios Katerelos. Automated tonal balance enhancement for audio mastering applications. In *Proceedings of the 134th AES Convention*, Rome, Italy, 2013. AES.
- [14] Enrique Perez Gonzales. *Advanced Automatic Mixing Tools for Music*. PhD thesis, Queen Mary University of London, London. UK, 30.09.2010.
- [15] Pedro D. Pestana and Joshua D. Reiss. A cross-adaptive dynamic spectral panning technique. In *Proceedings of the 17th International Conference on Digital Audio Effects (DAFx-14)*, Erlangen, Germany, 2014.
- [16] Jeffrey Scott and Youngmoo E. Kim. Analysis of acoustic features for automated multi-track mixing. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, Miami, USA, 2011.
- [17] Jeffrey Scott, Matthew Prockup, Erik M. Schmidt, and Youngmoo E. Kim. Automatic multi-track mixing using linear dynamical systems. In *Proceedings of the 8th Sound and Music Computing Conference (SMC)*, Padova, Italy, 2011.
- [18] Priyanka Shekar and Smith, III, Julius O. Modeling the harmonic exciter. In *Proceedings of the 135th AES Convention*, New York, USA, 2013.
- [19] U. Zölzer. *DAFX: Digital Audio Effects*. John Wiley & Sons Ltd., second edition, 2011.

AUTOMATIC TRANSCRIPTION OF ORNAMENTED IRISH TRADITIONAL FLUTE MUSIC USING HIDDEN MARKOV MODELS

Peter Jančovič¹

Münevver Köküer^{2,1}

Wrena Baptiste^{1,2}

¹ School of Electronic, Electrical & Systems Engineering, University of Birmingham, UK

² School of Digital Media Technology, Birmingham City University, UK

p.jancovic@bham.ac.uk, m.kokuer@bham.ac.uk

ABSTRACT

This paper presents an automatic system for note transcription of Irish traditional flute music containing ornamentation. This is a challenging problem due to the soft nature of onsets and short durations of ornaments. The proposed automatic transcription system is based on hidden Markov models, with separate models being built for notes and for single-note ornaments. Mel-frequency cepstral coefficients are employed to represent the acoustic signal. Different setups of parameters in feature extraction and acoustic modelling are explored. Experimental evaluations are performed on monophonic flute recordings from Grey Larsen's CD. The performance of the system is evaluated in terms of the transcription of notes as well as detection of onsets. It is demonstrated that the proposed system can achieve a very good note transcription and onset detection performance. Over 28% relative improvement in terms of the F -measure is achieved for onset detection in comparison to conventional onset detection methods based on signal energy and fundamental frequency.

1. INTRODUCTION

Automatic transcription of music is concerned with converting an acoustic signal into a symbolic representation that provides the information on individual notes played and possibly also other higher-level information about the structure of music. Over the last decade, there has been a considerable research interest in this field. Although most of the current research is devoted to polyphonic music transcription, transcription of monophonic music is still of interest due to existing large amount of real-world monophonic music of specific properties. This paper deals with the transcription of notes and detection of their onsets in monophonic flute recordings of Irish traditional music that contains ornamentation. Ornamentation is used extensively in Irish traditional music by players of all melody instruments. Ornaments are notes of a very short duration.

They are central to the style of the performer, adding to the liveliness and expression of the music.

A wide range of different approaches for automatic music transcription have been proposed. A variety of algorithms for estimating the fundamental frequency (F_0), e.g., [4, 10, 16], were employed in transcription of music, e.g., [1, 5, 10]. As the F_0 estimation may suffer from making octave errors, music transcription systems typically employ some way of temporal filtering or post-processing. The use of hidden Markov models (HMMs) for post-processing was presented in several works. In [2, 15] the sequence of pitch salience and onset strength or energy difference of adjacent signal frames were used as the input features to HMMs. In [6], the acoustic signal was first segmented by applying an onset detection algorithm and then HMM was used to track note candidates. Bayesian modeling that exploits knowledge of musical sound generation was proposed in [3, 9] and applied for piano transcription. Recently, several methods based on learning of a model / classifier of notes were presented, e.g., [7, 14]. In [14], a support vector machine classifier, trained on spectral features, was used to classify frame-level note instances and the classifier output was then temporally smoothed using a note level HMM to perform transcription of piano recordings. Modelling of a time-frequency representation of audio as a sum of basic elements representing the spectrum of a single note was presented in [7]. The transcription of ornamented Irish traditional flute music was investigated at the level of onset and ornament detection in [8, 11]. In both works, the presented ornament detection system was based on detecting onsets and using rules of musical ornamentation. An energy-based onset detection algorithm was employed in [8], while a comparison of two energy- and F_0 -based onset detection algorithms was performed in [11].

In this paper, we investigate an automatic transcription of ornamented Irish traditional flute music by employing hidden Markov models (HMMs). The proposed system is based on building an individual HMM for each note as well as for each ornament. This enables to model the differences in realisation of ornaments and notes and then detect ornaments whose fundamental frequency is close to the ornamented note. Music signal is represented as a sequence of Mel-frequency cepstral coefficients. Different parameter setups at various stages of the feature extraction and acoustic modelling are explored. Experimental evaluations



© Peter Jančovič, Münevver Köküer, Wrena Baptiste. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Peter Jančovič, Münevver Köküer, Wrena Baptiste. "Automatic transcription of ornamented Irish traditional flute music using hidden Markov models", 16th International Society for Music Information Retrieval Conference, 2015.

are performed using recordings of Irish traditional tunes played by flute from Grey Larsen’s CD [13]. Evaluations are presented for the task of onset detection and note transcription. Results are presented in terms of the precision, recall and F -measure. Onset detection evaluations are also compared to energy- and F_0 -based conventional onset detection algorithms. It is demonstrated that the proposed HMM-based transcription system achieves over 28% relative improvement in terms of the F -measure in onset detection task over conventional onset detection algorithms.

2. ORNAMENTED FLUTE MUSIC

2.1 Ornamentation in Irish traditional flute playing

Ornaments are used as embellishments in Irish traditional music [13]. They are notes of a very short duration, created through the use of special fingered articulations. They can be split into single- and multi-note ornaments. Single-note ornaments, namely ‘cut’ and ‘strike’, are pitch articulations. The ‘cut’ involves quickly lifting and replacing a finger from a tonehole, and corresponds to a higher note than the ornamented note. The ‘strike’ is performed by momentarily closing an open hole, and corresponds to a lower note than the ornamented note. Multi-note ornaments, namely ‘crann’, ‘roll’ and ‘shake’, are successive use of single-note ornaments. A schematic visualisation of the single- and multi-note ornaments in the time-frequency plane is given in Figure 1.

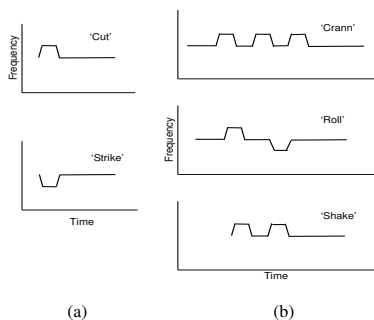


Figure 1. A schematic representation of single-note (a) and multi-note (b) ornaments in the time-frequency plane.

2.2 Annotation of the audio data

The audio signal was manually annotated by an experienced player of Irish traditional flute. The annotation provides segmentation of the audio signal, where each segment is characterised by the following: the time of onset, time of offset, type of segment, note identity (if applicable), and note frequency (if applicable). The type of segment may be one of the following: note, one of the types of single-note or multi-note ornaments, and breath. The note

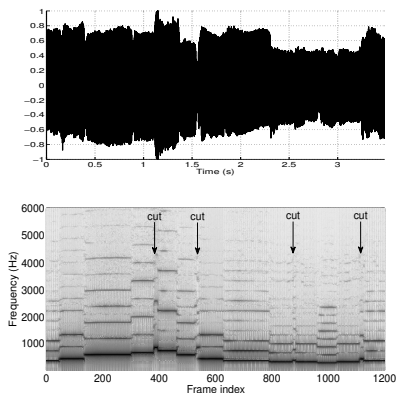


Figure 2. An extract from the tune ‘The Lonesome Jig’, depicting the waveform (top) and spectrogram (bottom).

frequency is initially estimated automatically but checked by the annotator, and, if needed, then corrected manually. Further information on the process of annotation of the audio recordings is presented in [12].

2.3 Data statistics

The flute music we are dealing with contains notes in the range from D4 to B5, i.e., with the fundamental frequency from 293 Hz to 987 Hz. Typically, only first few harmonics of the notes are having sufficient energy. An example of waveform and spectrogram is given in Figure 2. There are four instances of the ‘cut’ ornament indicated in the spectrogram at around frame indices 400, 540, 890 and 1130.

Based on the manual annotation, we examined the duration of the notes and single-note ornaments in our recordings. The obtained histograms, depicted in Figure 3, indicate that the duration of ornaments is considerably lower than that of notes. The mean duration of single-note ornaments is 63 ms, while it is 209 ms for notes. In 95% of cases, the duration of single-note ornaments is between 32 ms to 95 ms and of notes between 118 ms to 400 ms.

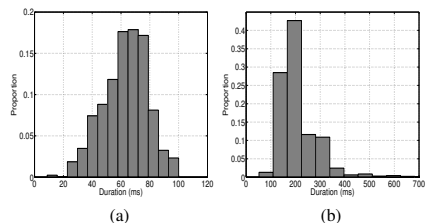


Figure 3. The distribution of the duration of single-note ornaments (a) and notes (b) in our recordings.

3. HMM-BASED NOTE TRANSCRIPTION AND ONSET DETECTION

This section presents the proposed HMM-based system for transcription of notes and detection of their onsets. We first describe the representation of the audio signal and then modelling using HMMs.

3.1 Feature representation

The acoustic signal is represented by a sequence of feature vectors, each vector capturing short-time spectral properties of the signal. Since we are dealing with unaccompanied music and in order to obtain lower-dimensional and less-correlated features, the signal is represented using Mel-frequency cepstral coefficients (MFCCs). MFCCs have been widely used in speech and audio processing. The steps involved in converting audio signal into MFCCs is depicted in Figure 4.

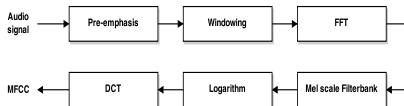


Figure 4. Processing steps used for converting the audio signal into a sequence of Mel-frequency cepstral features.

The signal is first segmented into overlapping frames. The frame length determines the temporal and frequency resolution. While longer frames allow for a finer frequency resolution, we are limited in the setting due to possibly very short duration of ornaments. Each signal frame is multiplied by Hamming window. The windowed frames are then zero padded and the Fourier transform is applied to provide the short-term Fourier spectrum. The short-time magnitude spectrum is passed through Mel-scale filter bank analysis and discrete cosine transform is applied on the logarithm of the filter-bank energies to provide MFCCs. In order to include information on local dynamics, MFCCs were appended with their temporal derivatives, referred to as the delta and acceleration features, which were calculated as in [17]. The values of the parameters within the processing steps need to reflect that we are dealing with ornamented flute music. As such, in our experimental evaluations, we explored various parameter setups.

3.2 Modelling

The model of each note is obtained by modelling the temporal evolution of feature vectors using a left-to-right (no skip allowed) hidden Markov model (HMM). We found that in addition to having an HMM for each note, it is essential to have also a separate HMM for each single-note ornament. This allows to deal with the fact that the realisation of single-note ornaments may not fully reach the

notional frequency of a note but rather be somewhere between two notes. In addition to this, we also create a model for breath and silence. These are used to model the taking a breath by the player and the initial and final silences in recordings, respectively. Overall, we have 42 models, consisting of 14 models for notes, 14 models for cuts, 11 models for strikes (strikes for some notes did not occur in our data) plus breath and silence. The state output probability density function (pdf) at each HMM state is modelled using a mixture of Gaussians. This allows for a more accurate modelling of variations in playing notes than using a single Gaussian distribution. Gaussian distributions with a diagonal covariance matrix are used due to computational reasons, as is typically done in speech and audio pattern processing. We explore the effect of using different number of HMM emitting states and Gaussian mixture components in the experimental section. The transcription system was built using the HTK [17].

3.2.1 Training

The initial values for the parameters of individual HMMs were estimated using isolated extracts from the audio signal, by applying several iterations of the Viterbi style training procedure. The isolated extracts were obtained based on the manual time-stamp annotation, i.e., onset and offset times. Further training of the models was then performed using several iterations of the Baum-Welch (aka forward-backward) algorithm. This uses continuous audio signal as input and requires only the sequence of notes/ornaments labels (i.e., no time-stamp). As such, this can eliminate the effect of possible errors in time-stamp annotation at borders of notes/ornaments on the trained models.

3.2.2 Recognition

To perform recognition, we need to construct a recognition network. This defines the allowed sequences of models (i.e., notes/ornaments). A network that closely reflects the knowledge of music could be employed. In this paper, we did not employ any such knowledge. We used a loop network that allows any note to follow any other note. We modified this slightly to reflect that an ornament model need to be followed by a note model. The network we used is depicted in Figure 5. As this network allows the same note to be subsequently repeated in the recognition output, we post-process the results such that the repetitions of the same note are considered to be a single instance of the note, for example, the original recognition output B4 D4 D4 A5 is considered to be B4 D4 A5. Note that a fixed probability value, aka word insertion penalty, can be associated with the transition from the end of one model to the start of the next model. This is useful in controlling the balance between the number of models being incorrectly inserted and deleted in the recognition result and we used it in our experimental evaluations.

Given a sequence of feature vectors, the Viterbi algorithm is used to find the optimal state sequence. This provides the sequence of recognised models as well as the start (and the end) times of each recognised model, i.e., the onset detection result.

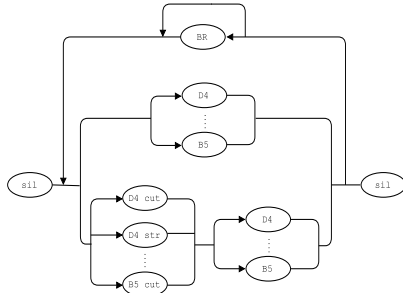


Figure 5. The recognition network used in the HMM-based note transcription system. The ellipses denote individual HMMs. Models of ornaments are denoted by note identity appended with either ‘cut’ or ‘str’, representing models for ‘cut’ and ‘strike’ ornaments, respectively. ‘BR’ denotes breath and ‘SIL’ silence models.

4. EXPERIMENTAL EVALUATIONS

4.1 Data description

Evaluations are performed using recordings of Irish traditional tunes and training exercises played by flute from Grey Larsen’s CD which accompanied his book “Essential Guide to Irish Flute and Tin Whistle” [13]. The tunes are between 16 sec and 1 min 22 sec long. All recordings are monophonic and are sampled at 44.1 kHz sampling frequency.

The collection consists of 19 tunes. The list of the tunes, with the number of notes and ornaments, is given in Table 1. In total, there are 3929 onsets, including notes and ornaments. Out of these there are 804 single-note ornaments (which includes also counts from parts of multi-note ornaments), consisting of 620 cuts and 184 strikes.

First evaluations are performed to demonstrate the effect of different parameter setups – due to computational reasons, these experiments use all files for training of models and also for testing. Final evaluations are performed using the leave-one-out cross-validation procedure, in which in turn 1 file is kept for testing and all the 18 remaining files are used for training. The results were accumulated over all files and then the evaluation measures calculated.

4.2 Evaluation measures

Performance of both the onset detection and note recognition is evaluated in terms of the precision (P), recall (R) and F -measure. The definition of these measures is the same as used in MIREX onset detection evaluations, specifically,

$$P = \frac{N_{tp}}{N_{tp} + N_{fp}}, R = \frac{N_{tp}}{N_{tp} + N_{fn}}, F = \frac{2PR}{P + R}.$$

In the case of onset detection, N_{tp} is the number of cor-

Tune Title	Number of			Time (sec.)
	Notes	Ornaments		
		Cut	Strike	
Study 5	55	16	0	20
Study 6	56	24	0	20
Study 11	76	20	0	26
Study 17	48	19	0	16
Study 22	127	0	28	47
Maids of Ardagh	98	28	5	32
Hardiman the ..	112	22	7	28
The Whinny Hills ..	117	34	6	30
The Frost is All ..	151	39	14	41
The Humours of ..	289	113	19	82
The Rose in the ..	152	33	13	39
Scotsman over ..	153	33	9	38
A Fig for a Kiss	105	27	9	28
Roaring Mary	176	42	22	44
The Mountain Road	105	20	6	25
The Shaskeen	181	52	23	42
Lady On The Island	118	21	1	21
The Lonesome Jig	153	27	0	46
The Drunken ..	185	50	22	43
Total	2457	620	184	668

Table 1. The list of tunes contained in the dataset, with the number of onsets and single-note ‘cut’ and ‘strike’ ornaments and duration of each tune.

rectly detected onsets and N_{fp} and N_{fn} is the number of inserted and deleted onsets, respectively. The onset detection is considered as correct when it is within ± 50 ms around the onset annotation.

In the case of note recognition, N_{tp} is the number of correctly recognised notes and N_{fp} and N_{fn} is the number of inserted and deleted notes, respectively.

4.3 Results for various parameter setups in feature extraction and modelling

This section explores the effect of different setups of parameters in the feature extraction and HMM-based modelling on the task of onset detection. A comparison with three conventional onset detection methods is also given.

4.3.1 Conventional onset detection algorithms

The conventional onset detection methods we employed to provide a comparison are: two methods which exploit the change of the signal amplitude over time, with processing performed in the temporal and spectral domain, and a method based on the fundamental frequency (F_0). The description of these methods, which we also used in our previous onset detection research, is provided in [11].

We performed extensive evaluations with different parameter values for each of the conventional onset detection methods. The best achieved performance for each of the methods is presented in Table 2. It can be seen that the F_0 -

based method performed better than each of the energy-based methods and achieved the F -measure of 91.2%.

Algorithm for onset detection	Evaluation performance (%)		
	Precision	Recall	F -measure
sig-energy (spectral)	94.8	85.6	89.9
sig-energy (temporal)	90.8	88.4	89.6
F_0	89.3	93.2	91.2

Table 2. Results of onset detection obtained using conventional onset detection methods.

4.3.2 HMM-based onset detection

Now we explore the performance of the HMM-based system when using various parameter setup in the feature extraction and acoustic modelling.

First, we compare results achieved by the HMM-based system when using the estimated F_0 with energy and the MFCCs (see the first row in Table 4 for the parameter setup) as the input features. Results are presented in Table 3. It can be seen that when using the estimated F_0 as input features to HMMs, the F -measure improved to 93.8%, from 91.2% that was achieved using the conventional F_0 -based method (as in Table 2). The performance of the HMM-based system improved considerably further to 96.7% when using MFCC features as input, instead of the estimated F_0 . As such, the use of HMMs driven with MFCC features provided over 60% error rate reduction over the best conventional method. The considerably better performance of the HMM-based system may be attributed to several factors. First, it is the statistical modelling of the temporal evolution of features. Second, the features used provide information about the spectral content. This is unlike the energy-based and F_0 -based conventional methods which accumulate the information from the entire signal bandwidth into a single detection function or into an F_0 estimate. Third, the use of HMM effectively incorporates smoothing of the frame-based decisions and imposes a minimum duration of notes and ornaments.

Features input to HMM	F -measure (%)
F_0 and energy, both with Δ and Δ^2	93.8
MFCC, both with Δ and Δ^2	96.7

Table 3. Results of the HMM-based onset detection when using an estimate of F_0 and MFCCs as input features.

Results obtained using different parameter setups in the MFCC feature extraction are presented in Table 4. The first line in the table presents the best parameter setup values and this is: bandwidth of 4 kHz, frame length of 12.5 ms, frame-shift of 5 ms, Mel-scale filter-bank with 21 channels, using 12 cepstral coefficients, and appending delta and acceleration coefficients (which were extracted using

Parameters in MFCC feature extraction	F -measure (%)
BW=4kHz, FrmL=12.5ms, FrmS=5ms, nFB=21, nCC=12, Δ and Δ^2	96.7
Bandwidth (BW)	95.5
Frame-length (FrmL)	95.4
	10 ms 95.9
	15 ms 96.3
	20 ms 96.3
	30 ms 95.7
Frame-shift (FrmS)	95.5
	7 ms 95.4
number of Mel filter-bank (nFB)	96.1
	25 96.5
Cepstral coefficients (nCC)	95.8
	14 96.6
Δ^2 coefficients	95.8
Δ and Δ^2 coefficients	91.6
	no

Table 4. Results of the HMM-based onset detection in terms of the F -measure obtained with different parameter setup in MFCC feature extraction.

the window of 3 and 2 signal frames, respectively). We now analyse the effect of each parameter – in each experiment, only one parameter is modified at a time in reference to the above best parameter setup. Let us start with varying the frequency bandwidth of the signal. This was performed at the stage of designing the Mel filter-banks. For the bandwidth of 6 kHz and 8 kHz, the number of filters was adjusted such that the lower 4 kHz was in all cases covered by 21 filters. It can be seen that the performance is similar when using the bandwidth of 6 kHz and 8 kHz but it improves considerably when the bandwidth is reduced to 4 kHz. This reflects, as we have also noticed in our visual inspection of spectrograms, that there is little signal content above 4 kHz in our flute recordings and as such the inclusion of the higher frequency range acts detrimentally to performance. This result may be used when analysing flute playing that contains accompaniments in higher frequencies or is recorded in live performances where other unwanted sounds may be present in higher frequencies. Next, results using different length of frames show that similar performance is achieved for lengths between 12 to 20 ms. The performance starts to decrease considerably when frames of 30 ms are used. This is due to the presence of ornaments, duration of which may be as short as 20 ms. In the case of frame shift, it can be seen that setting this to 3 ms or 7 ms considerably degrades the performance in comparison to the use of 5 ms shift. Varying the number of filter-bank channels from 17 to 25 has only relatively little effect, with performance being at the peak for 21 channels. The use of 12 or 14 cepstral coefficients provides very similar performance, while reducing this to 10 has quite negative effect. Finally, experiments when the delta and acceleration features, denoted by Δ and Δ^2 ,

respectively, are not included in feature representation are presented. Results show a large decrease in performance when neither delta nor acceleration features are used. This demonstrates the importance of incorporating information on local dynamics of the acoustic signal.

Next, we present the effect of different parameter setups in acoustic modelling. We vary the number of states and of mixture components of each state pdf for models of notes and ornaments. Results are presented in Table 5. The suitable range of values for the number of states of note and ornament models is determined based on the frame shift used in the feature extraction and statistics of the duration of the notes and ornaments. As such, we explored the range from 6 to 12 for notes and from 2 to 6 for ornaments. It can be seen that there is not much performance variation when using this range of values. In regard to the number of mixture components, it can be seen that it is useful to have at least 4 mixture components for note models, while even 2 mixture components seem sufficient for models of ornaments.

Parameters in acoustic modelling	<i>F</i> -measure (%)	
nStates for N / O / B: 8 / 4 / 8,	96.7	
nMix for N / O / B: 6 / 2 / 6		
nStates for notes	6	95.9
	10	96.1
	12	95.8
nStates for ornaments	2	95.8
	6	96.4
nMix for notes	2	95.6
	4	96.3
	8	96.6
nMix for ornaments	1	96.0
	4	96.7
	6	96.6

Table 5. Results of HMM-based onset detection in terms of the *F*-measure obtained with different parameter setup in acoustic modelling. N, O, and B stand for note, ornament and breath, respectively.

4.4 Results using the leave-one-out cross-validation

The final experimental evaluations are performed using the leave-one-out cross-validation. The feature extraction and acoustic modelling parameter setup that achieved best performance in previous section is used. The achieved results of onset detection and note identity recognition are presented in Table 6. It can be seen that very good performance is obtained for both tasks. The drop in onset detection performance in comparison to the results presented in the previous section is expected as the testing files have now not been seen during the training. Nevertheless, the performance is improved by over 28% relative over the conventional onset detection algorithms whose parameters were actually tuned based on both training and testing data.

	Evaluation performance (%)		
	Precision	Recall	<i>F</i> -measure
Onset detection	95.0	92.4	93.7
Note recognition	96.4	95.2	95.8

Table 6. Results of onset detection and note recognition obtained by the HMM-based system using the leave-one-out cross-validation procedure.

5. CONCLUSION AND FUTURE WORK

In this paper, we presented work on transcription of Irish traditional flute music containing ornamentation. The presented system is based on modelling of individual notes and ornaments using hidden Markov models. Acoustic signal is represented as a sequence of Mel frequency cepstral coefficients. A wide range of parameter setup values in both the feature extraction and acoustic modelling were explored. Experimental evaluations were performed using recordings of 19 Irish traditional flute tunes, containing in total 3929 onsets, out of which 804 corresponds to ornaments. Using the leave-one-out cross-validation procedure, the proposed HMM-based system achieved the *F*-measure of 93.7% in detecting onsets of ornaments and notes. This represents over 28% error rate reduction compared to conventional onset detectors whose parameters were even tuned to the testing data. The *F*-measure in the task of recognising the note identity was 95.8%.

There are several possible extensions of this work we are currently considering. First, the presented evaluations were performed using recordings from the same CD. We plan to perform evaluations on a range of recordings from several CDs in order to explore the capability of the system in dealing with variability due to different recording conditions, makes of flute instruments and performers. We will investigate techniques to compensate for such variability in order to improve robustness. Second, we plan to analyse the errors the automatic system makes in onset detection and note identity recognition tasks and reflect this in modifications to the system to further improve the performance. Then, the current HMM-based framework allows directly and in a probabilistic manner to incorporate musical knowledge on the sequences of notes used in flute music. Such knowledge could be obtained from musicologists and/or extracted automatically from annotations. Next, incorporation of an explicit duration modelling of notes and ornaments could help to reduce the number of falsely inserted and deleted onsets. Finally, we plan to expand the system to deal with recordings, in which the flute is accompanied by other instruments and/or singing.

6. ACKNOWLEDGEMENTS

This work was supported by the project ‘Characterising Stylistic Interpretations through Automated Analysis of Ornamentation in Irish Traditional Music Recordings’ under the ‘Transforming Musicology’ programme funded by the Arts and Humanities Research Council (UK).

7. REFERENCES

- [1] J. P. Bello, G. Monti, and M. Sandler. Techniques for automatic music transcription. In *Int. Symposium on Music Information Retrieval (ISMIR)*, Plymouth, USA, 2000.
- [2] E. Benetos and S. Dixon. Joint multi-pitch detection using harmonic envelope estimation for polyphonic music transcription. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1111–1123, October 2011.
- [3] A.T. Cemgil, H.J. Kappen, and D. Barber. A generative model for music transcription. *IEEE Trans. on Audio, Speech, and Language Processing*, 14(2):679–694, March 2006.
- [4] A. de Cheveigne and H. Kawahara. Yin, a fundamental frequency estimator for speech and music. *Journal of the Acoustical Society of America*, 111(4):1917–1930, April 2002.
- [5] K. Dressler. Extraction of the melody pitch contour from polyphonic audio. In *Int. Conf. on Music Information Retrieval (ISMIR)*, London, UK, 2005.
- [6] V. Emiya, R. Badeau, and B. David. Automatic transcription of piano music based on HMM tracking of jointly-estimated pitches. In *16th European Signal Processing Conf. (EUSIPCO)*, Lausanne, Switzerland, 2008.
- [7] B. Fuentes, R. Badeau, and G. Richard. Harmonic adaptive latent component analysis of audio and application to music transcription. *IEEE Trans. on Audio, Speech, and Language Processing*, 21(9):1854–1866, 2013.
- [8] M. Gainza and E. Coyle. Automating ornamentation transcription. In *IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, Honolulu, Hawaii, 2007.
- [9] S. Godsill and Manuel Davy. Bayesian harmonic models for musical pitch estimation and analysis. In *IEEE Int. Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, volume 2, pages II–1769–II–1772, May 2002.
- [10] A.P. Klapuri. Multiple fundamental frequency estimation based on harmonicity and spectral smoothness. *IEEE Transactions on Speech and Audio Processing*, 11(6):804–816, Nov 2003.
- [11] M. Kökür, P. Jančovič, I. Ali-MacLachlan, and C. Athwal. Automatic detection of single and multi-note ornaments in Irish traditional flute playing. In *Proc. of the 15th Int. Society for Music Information Retrieval Conf. (ISMIR)*, pages 15–20, Taipei, Taiwan, Nov 2014.
- [12] M. Kökür, D. Kearney, I. Ali-MacLachlan, P. Jančovič, and C. Athwal. Towards the creation of digital library content to study aspects of style in Irish traditional music. In *Proc. of the Int. Workshop on Digital Libraries for Musicology (DLFM)*, London, 2014.
- [13] G. Larsen. *The Essential Guide to Irish Flute and Tin Whistle*. Mel Bay Publications, Pacific, Missouri, USA, 2003.
- [14] G. E. Poliner and D. Ellis. A discriminative model for polyphonic piano transcription. *EURASIP Journal on Advances in Signal Processing*, 2007(1):048317, 2007.
- [15] M. P. Ryyänen and A. P. Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32(3):72–86, 2008.
- [16] J. Salamon and E. Gómez. Melody extraction from polyphonic music signals using pitch contour characteristics. *IEEE Trans. on Audio, Speech, and Language Processing*, 20(6):1759–1770, 2012.
- [17] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland. *The HTK Book*. V2.2. 1999.

TOWARDS MUSIC IMAGERY INFORMATION RETRIEVAL: INTRODUCING THE OPENMIIR DATASET OF EEG RECORDINGS FROM MUSIC PERCEPTION AND IMAGINATION

Sebastian Stober, Avital Sternin, Adrian M. Owen and Jessica A. Grahn

Brain and Mind Institute, Department of Psychology, Western University, London, ON, Canada

{sstober, asternin, adrian.owen, jgrahn}@uwo.ca

ABSTRACT

Music imagery information retrieval (MIIR) systems may one day be able to recognize a song from only our thoughts. As a step towards such technology, we are presenting a public domain dataset of electroencephalography (EEG) recordings taken during music perception and imagination. We acquired this data during an ongoing study that so far comprises 10 subjects listening to and imagining 12 short music fragments – each 7–16s long – taken from well-known pieces. These stimuli were selected from different genres and systematically vary along musical dimensions such as meter, tempo and the presence of lyrics. This way, various retrieval scenarios can be addressed and the success of classifying based on specific dimensions can be tested. The dataset is aimed to enable music information retrieval researchers interested in these new MIIR challenges to easily test and adapt their existing approaches for music analysis like fingerprinting, beat tracking, or tempo estimation on EEG data.

1. INTRODUCTION

We all imagine music in our everyday lives. Individuals can imagine themselves producing music, imagine listening to others produce music, or simply “hear” the music in their heads. Music imagination is used by musicians to memorize music pieces and anyone who has ever had an “ear-worm” – a tune stuck in their head – has experienced imagining music. Recent research also suggests that it might one day be possible to retrieve a music piece from a database by just thinking of it.

As already motivated in [29], music imagery information retrieval (MIIR) – i.e., retrieving music by imagination – has the potential to overcome the query expressivity bottleneck of current music information retrieval (MIR) systems, which require their users to somehow imitate the desired song through singing, humming, or beat-boxing [31] or to describe it using tags, meta-data, or lyrics fragments. Furthermore, music imagery appears to be a very promising means for driving brain-computer in-

terfaces (BCIs) that use electroencephalography (EEG) – a popular non-invasive neuroimaging technique that relies on electrodes placed on the scalp to measure the electrical activity of the brain. For instance, Schaefer et al. [23] argue that “*music is especially suitable to use here as (externally or internally generated) stimulus material, since it unfolds over time, and EEG is especially precise in measuring the timing of a response.*” This allows us to exploit temporal characteristics of the signal such as rhythmic information.

Still, EEG data is generally very noisy and thus extracting relevant information can be challenging. This calls for sophisticated signal processing techniques as they have emerged in the field of MIR within the last decade. However, MIR researchers with the potential expertise to analyze music imagery data usually do not have access to the required equipment to acquire the necessary data for MIIR experiments in the first place.¹ In order to remove this substantial hurdle and encourage the MIR community to try their methods in this emerging interdisciplinary field, we are introducing the *OpenMIIR* dataset.

In the following sections, we will review closely related work in Section 2, describe our approach for data acquisition (Section 3) and basic processing (Section 4), and outline further steps in Section 5.

2. RELATED WORK

Retrieval based on brain wave recordings is still a very young and largely unexplored domain. A recent review of neuroimaging methods for MIR that also covers techniques different from EEG is given in [14]. EEG signals have been used to measure emotions induced by music perception [1, 16] and to distinguish perceived rhythmic stimuli [28]. It has been shown that oscillatory neural activity in the gamma frequency band (20-60 Hz) is sensitive to accented tones in a rhythmic sequence [27]. Oscillations in the beta band (20-30 Hz) entrain to rhythmic sequences [2, 17] and increase in anticipation of strong tones in a non-isochronous, rhythmic sequence [5, 6, 13]. The magnitude of steady state evoked potentials (SSEPs), which reflect neural oscillations entrained to the stimulus, changes when subjects hear rhythmic sequences for frequencies related to the metrical structure of the rhythm. This is a sign of entrainment to beat and meter [19, 20]. EEG studies have further shown that perturbations

¹For instance, the Biosemi EEG system used here costs several thousand dollars. Consumer-level EEG devices with a much lower price have become available recently but it is still open whether their measuring precision and resolution is sufficient for MIIR research.



© Sebastian Stober, Avital Sternin, Adrian M. Owen and Jessica A. Grahn.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Sebastian Stober, Avital Sternin, Adrian M. Owen and Jessica A. Grahn. “Towards Music Imagery Information Retrieval: Introducing the OpenMIIR Dataset of EEG Recordings from Music Perception and Imagination”, 16th International Society for Music Information Retrieval Conference, 2015.

of the rhythmic pattern lead to distinguishable event-related potentials (ERPs)² [7]. This effect appears to be independent of the listener's level of musical proficiency. Furthermore, Vlek et al. [32] showed that imagined auditory accents imposed on top of a steady metronome click can be recognized from EEG.

EEG has also been successfully used to distinguish perceived melodies. In a study by Schaefer et al. [26], 10 participants listened to 7 short melody clips with a length between 3.26s and 4.36s. For single-trial classification, each stimulus was presented 140 times in randomized back-to-back sequences of all stimuli. Using a quadratically regularized linear logistic-regression classifier with 10-fold cross-validation, they were able to successfully classify the ERPs of single trials. Within subjects, the accuracy varied between 25% and 70%. Applying the same classification scheme across participants, they obtained between 35% and 53% accuracy. In a further analysis, they combined all trials from all subjects and stimuli into a grand average ERP. Using singular-value decomposition, they obtained a fronto-central component that explained 23% of the total signal variance. The time courses corresponding to this component showed significant differences between stimuli that were strong enough to allow cross-participant classification. Furthermore, a correlation with the stimulus envelopes of up to 0.48 was observed with the highest value over all stimuli at a time lag of 70–100ms.

fMRI studies [10, 11] have shown that similar brain structures and processes are involved during music perception and imagination. As Hubbard concludes in his recent review of the literature on auditory imagery, “*auditory imagery preserves many structural and temporal properties of auditory stimuli*” and “*involves many of the same brain areas as auditory perception*” [12]. This is also underlined by Schaefer [23, p. 142] whose “*most important conclusion is that there is a substantial amount of overlap between the two tasks [music perception and imagination], and that ‘internally’ creating a perceptual experience uses functionalities of ‘normal’ perception.*” Thus, brain signals recorded while listening to a music piece could serve as reference data. The data could be used in a retrieval system to detect salient elements expected during imagination. A recent meta-analysis [25] summarized evidence that EEG is capable of detecting brain activity during the imagination of music. Most notably, encouraging preliminary results for recognizing imagined music fragments from EEG recordings were reported in [24] in which 4 out of 8 participants produced imagery that was classifiable (in a binary comparison) with an accuracy between 70% and 90% after 11 trials.

Another closely related field of research is the reconstruction of auditory stimuli from EEG recordings. Deng et al. [3] observed that EEG recorded during listening to natural speech contains traces of the speech amplitude envelope. They used independent component analysis (ICA) and a source localization technique to enhance the strength of this signal and successfully identify heard sentences. Applying their technique to imagined speech, they reported statistically significant single-sentence classification performance for 2 of 8 subjects with better performance when several sentences were combined for

a longer trial duration.

Recently, O’Sullivan et al. [21] proposed a method for decoding attentional selection in a cocktail party environment from single-trial EEG recordings approximately one minute long. In their experiment, 40 subjects were presented with 2 classic works of fiction at the same time – each one to a different ear – for 30 trials. To determine which of the 2 stimuli a subject attended to, they reconstructed both stimulus envelopes from the recorded EEG. To this end, they trained two different decoders per trial using a linear regression approach – one to reconstruct the attended stimulus and the other to reconstruct the unattended one. This resulted in 60 decoders per subject. These decoders were then averaged in a leave-one-out cross-validation scheme. During testing, each decoder would predict the stimulus with the best reconstruction from the EEG using the Pearson correlation of the envelopes as measure of quality. Using subject-specific decoders averaged from 29 training trials, the prediction of the attended stimulus decoder was correct for 89% of the trials whereas the mean accuracy of the unattended stimulus decoder was 78.9%. Alternatively, using a grand-average decoding method that combined the decoders from every other subject and every other trial, they obtained a mean accuracy of 82% and 75% respectively.

3. STUDY DESCRIPTION

This section provides details about the study that was conducted to collect the data released in the OpenMIIR dataset. The study consisted of two portions. We first collected information about the participants using questionnaires and behavioral testing (Section 3.1) and then ran the actual EEG experiment (Section 3.2) with those participants matching our inclusion criteria. The 12 music stimuli used in this experiment are described in Section 3.3.

3.1 Questionnaires and Behavioral Testing

14 participants were recruited using approved posters at the University of Western Ontario. We collected information about the participants’ previous music experience, their ability to imagine sounds, and information about musical sophistication using an adapted version of the widely used Goldsmith’s Musical Sophistication Index (G-MSI) [18] combined with an adapted clarity of auditory imagination scale [33]. Questions from the perceptual abilities and musical training subscales of the G-MSI were used to identify individual differences in these areas. For the clarity of auditory imagery scale, participants had to self-report their ability to clearly hear sounds in their head. Our version of this scale added five music-related items to five items from the original scale.

We also had participants complete a beat tapping and a stimuli familiarity task. Participants listened to each stimulus and were asked to tap along with the music on the table top. The experimenter then rated their tapping ability on a scale from 1 (difficult to assess) to 3 (tapping done properly). After listening to each stimulus participants rated their familiarity with the stimuli on a scale from 1 (unfamiliar) to 3 (very familiar). To participate in the EEG portion of the study, the participants had to receive a score of at least 90% on our beat tapping task.

² A description of how event-related potentials (ERPs) are computed and some examples are provided in Section 4.

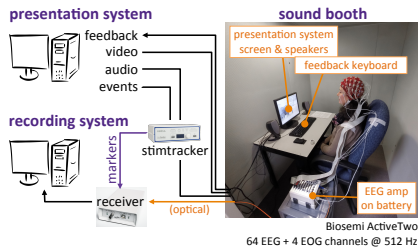


Figure 1. Setup for the EEG experiment. The presentation and recording systems were placed outside to reduce the impact of electrical line noise that could be picked up by the EEG amplifier.

Participants received scores from 75%–100% with an average score of 96%. Furthermore, they needed to receive a score of at least 80% on our stimuli familiarity task. Participants received scores from 71%–100% with an average score 87%. These requirements resulted in rejecting 4 participants. This left 10 participants (3 male), aged 19–36, with normal hearing and no history of brain injury. These 10 participants had an average tapping score of 98% and an average familiarity score of 92%. Eight participants had formal musical training (1–10 years), and four of those participants played instruments regularly at the time of data collection. After the experiment, we asked participants the method they used to imagine music. The participants were split evenly between imagining themselves producing the music (singing or humming) and simply “hearing the music in [their] head.”

3.2 EEG Recording

For the EEG portion of the study, the 10 participants were seated in an audiometric room (Eckel model CL-13) and connected to a BioSemi Active-Two system recording 64+2 EEG channels at 512 Hz as shown in Figure 1. Horizontal and vertical EOG channels were used to record eye movements. We also recorded the left and right mastoid channel as EEG reference signals. Due to an oversight, the mastoid data was not collected for the first 5 subjects. The presented audio was routed through a Cedrus StimTracker connected to the EEG receiver, which allowed a high-precision synchronization (<0.05 ms) of the stimulus onsets with the EEG data. The experiment was programmed and presented using PsychToolbox run in Matlab 2014a. A computer monitor displayed the instructions and fixation cross for the participants to focus on during the trials to reduce eye movements. The stimuli and cue clicks were played through speakers at a comfortable volume that was kept constant across participants. Headphones were not used because pilot participants reported headphones caused them to hear their heartbeat which interfered with the imagination portion of the experiment.

The EEG experiment was divided into 2 parts with 5 blocks each as illustrated in Figure 2. A single block comprised of all

Table 1. Information about the tempo, meter and length of the stimuli (without cue clicks) used in this study.

ID	Name	Meter	Length	Tempo
1	Chim Chim Cheree (lyrics)	3/4	13.3s	212 BPM
2	Take Me Out to the Ballgame (lyrics)	3/4	7.7s	189 BPM
3	Jingle Bells (lyrics)	4/4	9.7s	200 BPM
4	Mary Had a Little Lamb (lyrics)	4/4	11.6s	160 BPM
11	Chim Chim Cheree	3/4	13.5s	212 BPM
12	Take Me Out to the Ballgame	3/4	7.7s	189 BPM
13	Jingle Bells	4/4	9.0s	200 BPM
14	Mary Had a Little Lamb	4/4	12.2s	160 BPM
21	Emperor Waltz	3/4	8.3s	178 BPM
22	Hedwig’s Theme (Harry Potter)	3/4	16.0s	166 BPM
23	Imperial March (Star Wars Theme)	4/4	9.2s	104 BPM
24	Eine Kleine Nachtmusik	4/4	6.9s	140 BPM
mean			10.4s	176 BPM

12 stimuli in randomized order. Between blocks, participants could take breaks at their own pace. We recorded EEG in 4 conditions:

1. Stimulus perception preceded by cue clicks
2. Stimulus imagination preceded by cue clicks
3. Stimulus imagination without cue clicks
4. Stimulus imagination without cue clicks, with feedback

The goal was to use the cue to align trials of the same stimulus collected under conditions 1 and 2. Lining up the trials allows us to directly compare the perception and imagination of music and to identify overlapping features in the data. Conditions 3 and 4 simulate a more realistic query scenario during which the system does not have prior information about the tempo and meter of the imagined stimulus. These two conditions were identical except for the trial context. While the condition 1–3 trials were recorded directly back-to-back within the first part of the experiment, all condition 4 trials were recorded separately in the second part, without any cue clicks or tempo priming by prior presentation of the stimulus. After each condition 4 trial, participants provided feedback by pressing one of two buttons indicating on whether or not they felt they had imagined the stimulus correctly. In total, 240 trials (12 stimuli x 4 conditions x 5 blocks) were recorded per subject. The event markers recorded in the raw EEG comprise:

- Trial labels (as a concatenation of stimulus ID and condition) at the beginning of each trial
- Exact audio onsets for the first cue click of each trial in conditions 1 and 2 (detected by the Stimtracker)
- Subject feedback for the condition 4 trials (separate event IDs for positive and negative feedback)

3.3 Stimuli

Table 1 shows an overview of the stimuli used in the study. This selection represents a tradeoff between exploration and exploitation of the stimulus space. As music has many facets, there are naturally many possible dimensions in which music pieces may vary. Obviously, only a limited subspace could be explored with any given set of stimuli. This had to be balanced against the number of trials that could be recorded for each stimulus (exploitation) within a given time limit of 2 hours for a single recording session (including fitting the EEG equipment).

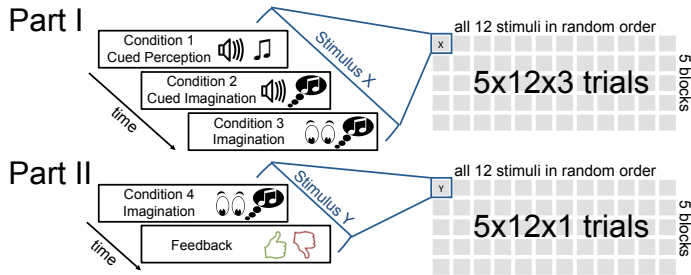


Figure 2. Illustration of the design for the EEG portion of the study.

Based on the findings from related studies (c.f. Section 2), we primarily focused on the rhythm/meter and tempo dimensions. Consequently, the set of stimuli was evenly divided into pieces with 3/4 and 4/4 meter, i.e. two very distinct rhythmic “feels.” The tempo spanned a range between 104 and 212 beats per minute (BPM). Furthermore, we were also interested in whether the presence of lyrics would improve the recognizability of the stimuli. Hence, we divided the stimulus set into 3 equally sized groups:

- 4 recordings of songs with lyrics (1–4),
- 4 recordings of the same songs without lyrics (11–14), and
- 4 instrumental pieces (21–24).

The pairs of recordings for the same song with and without lyrics were tempo-matched by pre-selection and subsequent fine adjustment using the time-stretching function of Audacity.³ Due to minor differences in tempo between pairs of stimuli with and without lyrics, the tempo of the stimuli had to be slightly modified after the first five participants.

All stimuli were considered to be well-known pieces in the North-American cultural context. They were normalized in volume and kept as similar in length as possible with care taken to ensure that they all contained complete musical phrases starting from the beginning of the piece. Each stimulus started with approximately two seconds of clicks (1 or 2 bars) as an auditory cue to the tempo and onset of the music. The clicks began to fade out at the 1s-mark within the cue and stopped at the onset of the music.

3.4 Data and Code Sharing

With the explicit consent of all participants and the approval of the ethics board at the University of Western Ontario, the data collected in this study are released as OpenMIIR dataset⁴ under the *Open Data Commons Public Domain Dedication and License (PDDL)*.⁵ This comprises the anonymized answers from the questionnaires, the behavioral scores, the subjects’ feedback for the trials in condition 4 and the raw EEG and EOG data of all trials at the original sample rate of 512 Hz. This amounts to approximately 700 MB of data per subject.

³ <http://web.audacityteam.org/>

⁴ <https://github.com/sstober/openmiir>

⁵ <http://opendatacommons.org/licenses/pddl>

Raw data are shared in the FIF format used by MNE [9], which can easily be converted to the MAT format of Matlab.

Additionally, the Matlab code and the stimuli for running the study are made available as well as the python code for cleaning and processing the raw EEG data as described in Section 4. The python code uses the libraries MNE-Python [8] and deepthought⁶, which are both published as open-source under the 3-clause BSD license.⁷

This approach ensures accessibility and reproducibility. Researchers have the possibility to just apply their methods on the already pre-processed data or change any step in the pre-processing pipeline according to their needs. No proprietary software is required for working with the data. The wiki on the dataset website can be used to share code, ideas and results related to the dataset.

4. BASIC EEG PROCESSING

This section describes basic EEG processing techniques that may serve as a basis for the application of more sophisticated analysis methods. More examples are linked in the wiki on the dataset website.

4.1 EEG Data Cleaning

EEG recordings are usually very noisy. They contain artifacts caused by muscle activity such as eye blinking as well as possible drifts in the impedance of the individual electrodes over the course of a recording. Furthermore, the recording equipment is very sensitive and easily picks up interferences such as electrical line noise from the surroundings. The following common-practice pre-processing steps were applied to remove unwanted artifacts.

The raw EEG and EOG data were processed using the MNE-Python toolbox. The data was first visually inspected for artifacts. For one subject (P05), we identified several episodes of strong movement artifacts during trials. Hence, these particular data need to be treated with care when used for analysis – possibly picking only specific trials without artifacts. The bad

⁶ <https://github.com/sstober/deepthought>

⁷ <http://opensource.org/licenses/BSD-3-Clause>

trials might however still be used for testing the robustness of analysis techniques.

For recordings with additional mastoid channels, the EEG data was re-referenced by subtracting the mean mastoid signal [30]. We then removed and interpolated bad EEG channels identified by manual visual inspection. For interpolation, the spherical splines method described in [22] was applied. The number of bad channels in a single recording session varied between 0 and 3. The data were then filtered with an fit-bandpass, keeping a frequency range between 0.5 and 30 Hz. This also removed any slow signal drift in the EEG. Afterwards, we down-sampled to a sampling rate of 64 Hz. To remove artifacts caused by eye blinks, we computed independent components using extended Infomax ICA [15] and semi-automatically removed components that had a high correlation with the EOG channels. Finally, the 64 EEG channels were reconstructed from the remaining independent components without reducing dimensionality.

4.2 Grand Average Trial ERPs

A common approach to EEG analysis is through the use of event-related potentials (ERPs). An ERP is an electrophysiological response that occurs as a direct result of a stimulus. Raw EEG data is full of unwanted signals. In order to extract the signal of interest from the noise, participants are presented with the same stimulus many times. The brain's response to the stimulus remains constant while the noise changes. The consistent brain response becomes apparent when the signals from the multiple stimulus presentations are averaged together and the random noise is averaged to zero. In order to identify common brain response patterns across subjects, grand average ERPs are computed by averaging the ERPs of different subjects.

The size and the timing of peaks in the ERP waveform provide information about the brain processes that occur in response to the presented stimulus. By performing a principle component analysis (PCA), information regarding the spatial features of these processes can be obtained.

As proposed in [26], we computed grand average ERPs by aggregating over all trials (excluding the cue clicks) of the same stimulus from all subjects except P05 (due to the movement artifacts). In their experiment, Schaefer et al. [26] used very short stimuli allowing each stimulus to be repeated many times. They averaged across hundreds of short (3.26s) trials, concatenated the obtained grand average ERPs and then applied PCA, which resulted in clearly defined spatial components. We had fewer repetitions of our stimuli. Therefore, to preserve as much data as possible, we used the full length of the trials as opposed to the first 3.26 seconds. We then concatenated the grand average ERPs and applied a PCA, which resulted in principal components with poorly defined spatial features as shown in Figure 3 (A and B). As an alternative, we performed a PCA on the concatenated raw trials without first calculating an average across trials. This approach produced clearly defined spatial components shown in Figure 3 (C and D). Components 2 to 4 are similar to those described in [26]. Except for their (arbitrary) polarity, the components are very similar across the two conditions, which may be indicative of similar processes being involved in both perception and imagination of music as

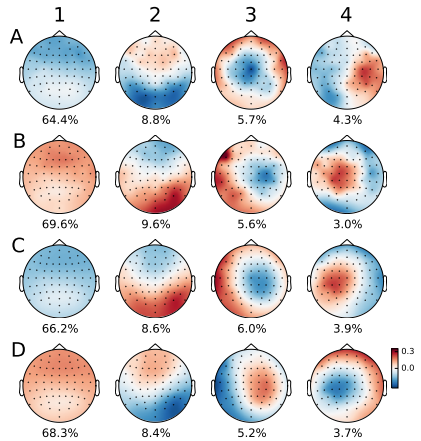


Figure 3. Topographic visualization of the top 4 principle components with percentage of the explained signal variance. Channel positions in the 64-channel EEG layout are shown as dots. Colors are interpolated based on the channel weights. The PCA was computed on **A:** the grand average ERPs of all perception trials, **B:** the grand average ERPs of all cued imagination trials, **C:** the concatenated perception trials, **D:** the concatenated cued imagination trials.

described in [11, 25].

Schaefer et al. [26] were able to use the unique time course of the component responsible for the most variance to differentiate between stimuli. Analyzing the signals corresponding to the principle components, we have not yet been able to reproduce a significant stimulus classification accuracy. This could be caused by our much smaller number of trials, which are also substantially longer than those used by [26]. Furthermore, the cross-correlation between the stimulus envelopes and the component waveforms were much lower (often below 0.1) than reported in [26].

4.3 Grand Average Beat ERPs

In the previous section, we computed ERPs based on the trial onsets. Similarly, it is also possible to analyze beat events. Using the dynamic beat tracker [4] provided by the *librosa*⁸ library, we obtained beat annotations for all beats within the audio stimuli. To this end, the beat tracker was initialized with the known tempo of each stimulus. The quality of the automatic annotations was verified through sonification.

Knowing the beat positions allows to analyze the respective EEG segments in the perception condition. For this analysis, the EEG data was additionally filtered with a low-pass at 8 Hz to remove alpha band activity (8–12 Hz). Figure 4 shows

⁸<https://github.com/bmcfee/librosa>

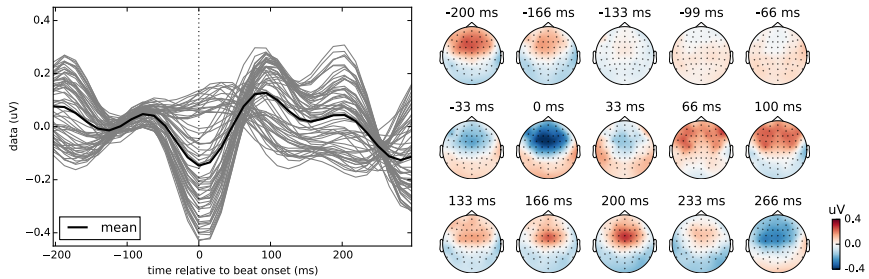


Figure 4. Grand average beat ERP for the perception trials (16515 beats). All times are relative to the beat onset. Left: Individual channels and mean over time. Right: Topographic visualization for discrete time points (equally spaced at 1/30s interval).

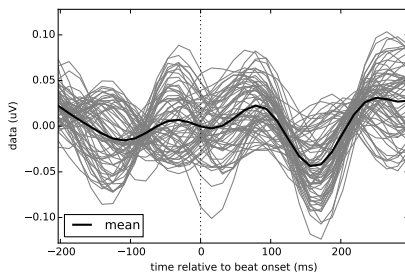


Figure 5. Grand average beat ERP for the cued imagination trials (16515 beats). All times are relative to the beat onset. Note the difference in amplitude compared to Figure 4.

the grand average ERP for all beats except the cue clicks⁹ in all perception trials of all subjects except P05. Here we considered epochs, i.e., EEG segments of interest, from 200 ms before until 300 ms after each beat marker. Before averaging into the ERP, we applied a baseline correction of each epoch by subtracting the signal mean computed from the 200 ms sub-segment before the beat marker.

The ERP has a negative dip that coincides with the beat onset time at 0 ms. Any auditory processing related to the beat would occur much later. A possible explanation is that the dip is caused by the anticipation of the beat. However, this requires further investigation. There might be potential to use this effect as the basis for an MIIR beat or tempo tracker. For comparison, the respective grand average ERP for the cued imagination trials is shown in Figure 5. This ERP looks very different from the one for the perception conditions. Most notably the amplitude scale is very low. This outcome was probably caused by the imprecise time locking. In order to compute meaningful ERPs, the precise event times (beat onsets) need to be known. However, small tempo variations during imagination are very likely and thus the beat onsets are most likely not exact.

⁹ Cue clicks were excluded because these isolated auditory events elicit a different brain response than beats embedded into a stream of music.

5. CONCLUSIONS AND OUTLOOK

We have introduced OpenMIIR – an open EEG dataset intended to enable MIR researchers to venture into the domain of music imagery and develop novel methods without the need for special EEG equipment. We plan to add new EEG recordings with further subjects to the dataset and possibly adapt the experimental settings as we learn more about the problem. In our first experiments using this dataset, we were able to partly reproduce the identification of overlapping components between music perception and imagination as reported earlier.

Will it one day be possible to just think of a song and the music player will start its playback? If this could be achieved, it would require the intense interdisciplinary collaboration between MIR researchers and neuroscientists. We hope that the OpenMIIR dataset will facilitate such a collaboration and contribute to new developments in this emerging field for research. **Acknowledgments:** This work has been supported by a fellowship within the Postdoc-Program of the German Academic Exchange Service (DAAD), the Canada Excellence Research Chairs (CERC) Program, an National Sciences and Engineering Research Council (NSERC) Discovery Grant, an Ontario Early Researcher Award, and the James S. McDonnell Foundation. The authors would further like to thank the study participants, and the anonymous ISMIR reviewers for the constructive feedback on the paper.

6. REFERENCES

- [1] R. Cabredo, R. S. Legaspi, P. S. Inventado, and M. Numao. An Emotion Model for Music Using Brain Waves. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR'12)*, pages 265–270, 2012.
- [2] L. K. Cirelli, D. Bosnyak, F. C. Manning, C. Spinelli, C. Marie, T. Fujioka, A. Ghahremani, and L. J. Trainor. Beat-induced fluctuations in auditory cortical beta-band activity: Using EEG to measure age-related changes. *Frontiers in Psychology*, 5(Jul):1–9, 2014.
- [3] S. Deng, R. Srinivasan, and M. D'Zmura. Cortical signatures of heard and imagined speech envelopes. Technical report, DTIC, 2013.

- [4] D. P. W. Ellis. Beat Tracking by Dynamic Programming. *Journal of New Music Research*, 36(1):51–60, 2007.
- [5] T. Fujioka, L. J. Trainor, E. W. Large, and B. Ross. Beta and gamma rhythms in human auditory cortex during musical beat processing. *Annals of the New York Academy of Sciences*, 1169:89–92, 2009.
- [6] T. Fujioka, L. J. Trainor, E. W. Large, and B. Ross. Internalized Timing of Isochronous Sounds Is Represented in Neuromagnetic Beta Oscillations. *Journal of Neuroscience*, 32(5):1791–1802, 2012.
- [7] E. Geiser, E. Ziegler, L. Jancke, and M. Meyer. Early electrophysiological correlates of meter and rhythm processing in music perception. *Cortex*, 45(1):93–102, 2009.
- [8] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, R. Goj, M. Jas, T. Brooks, L. Parkkonen, and M. Hämäläinen. MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 7, 2013.
- [9] A. Gramfort, M. Luessi, E. Larson, D. A. Engemann, D. Strohmeier, C. Brodbeck, L. Parkkonen, and M. S. Hämäläinen. MNE software for processing MEG and EEG data. *NeuroImage*, 86(0):446–460, 2014.
- [10] A. R. Halpern, R. J. Zatorre, M. Bouffard, and J. A. Johnson. Behavioral and neural correlates of perceived and imagined musical timbre. *Neuropsychologia*, 42(9):1281–92, 2004.
- [11] S. Herholz, A. Halpern, and R. Zatorre. Neuronal correlates of perception, imagery, and memory for familiar tunes. *Journal of cognitive neuroscience*, 24(6):1382–97, 2012.
- [12] T. L. Hubbard. Auditory imagery: empirical findings. *Psychological Bulletin*, 136(2):302–329, 2010.
- [13] J. R. Iversen, B. H. Repp, and A. D. Patel. Top-down control of rhythm perception modulates early auditory responses. *Annals of the New York Academy of Sciences*, 1169:58–73, 2009.
- [14] B. Kaneshiro and J. P. Dmochowski. Neuroimaging methods for music information retrieval: Current findings and future prospects. In *Proceedings of the 16th International Society for Music Information Retrieval Conference (ISMIR'15)*, 2015.
- [15] T.-W. Lee, M. Girolami, and T. J. Sejnowski. Independent Component Analysis Using an Extended Infomax Algorithm for Mixed Subgaussian and Supergaussian Sources. *Neural Computation*, 11(2):417–441, 1999.
- [16] Y.-P. Lin, T.-P. Jung, and J.-H. Chen. EEG dynamics during music appreciation. In *Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC'09)*, pages 5316–5319, 2009.
- [17] H. Merchant, J. Grahn, L. J. Trainor, M. Rohrmeier, and W. T. Fitch. Finding a beat: a neural perspective across humans and non-human primates. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 2015.
- [18] D. Müllensiefen, B. Gingras, J. Musil, and L. Stewart. The Musicality of Non-Musicians: An Index for Assessing Musical Sophistication in the General Population. *PLoS ONE*, 9(2), 2014.
- [19] S. Nozaradan, I. Peretz, M. Missal, and A. Mouraux. Tagging the neuronal entrainment to beat and meter. *The Journal of Neuroscience*, 31(28):10234–10240, 2011.
- [20] S. Nozaradan, I. Peretz, and A. Mouraux. Selective Neuronal Entrainment to the Beat and Meter Embedded in a Musical Rhythm. *The Journal of Neuroscience*, 32(49):17572–17581, 2012.
- [21] J. A. O’Sullivan, A. J. Power, N. Mesgarani, S. Rajaram, J. J. Foxe, B. G. Shinn-Cunningham, M. Slaney, S. A. Shamma, and E. C. Lalor. Attentional Selection in a Cocktail Party Environment Can Be Decoded from Single-Trial EEG. *Cerebral Cortex*, 25:1697–1706, 2015.
- [22] F. Perrin, J. Pernier, O. Bertrand, and J. F. Echallier. Spherical splines for scalp potential and current density mapping. *Electroencephalography and Clinical Neurophysiology*, 72(2):184–187, 1989.
- [23] R. Schaefer. *Measuring the mind’s ear EEG of music imagery*. PhD thesis, Radboud University Nijmegen, 2011.
- [24] R. Schaefer, Y. Blokländ, J. Farquhar, and P. Desain. Single trial classification of perceived and imagined music from EEG. In *Proceedings of the 2009 Berlin BCI Workshop*. 2009.
- [25] R. S. Schaefer, P. Desain, and J. Farquhar. Shared processing of perception and imagery of music in decomposed EEG. *NeuroImage*, 70:317–326, 2013.
- [26] R. S. Schaefer, J. Farquhar, Y. Blokländ, M. Sadakata, and P. Desain. Name that tune: Decoding music from the listening brain. *NeuroImage*, 56(2):843–849, 2011.
- [27] J. S. Snyder and E. W. Large. Gamma-band activity reflects the metric structure of rhythmic tone sequences. *Cognitive Brain Research*, 24:117–126, 2005.
- [28] S. Stober, D. J. Cameron, and J. A. Grahn. Using convolutional neural networks to recognize rhythm stimuli from electroencephalography recordings. In *Advances in Neural Information Processing Systems 27 (NIPS'14)*, pages 1449–1457, 2014.
- [29] S. Stober and J. Thompson. Music imagery information retrieval: Bringing the song on your mind back to your ears. In *13th International Conference on Music Information Retrieval (ISMIR'12) - Late-Breaking & Demo Papers*, 2012.
- [30] M. Teplan. Fundamentals of EEG measurement. *Measurement science review*, 2(2):1–11, 2002.
- [31] G. Tzanetakis, A. Kapur, and M. Benning. Query-by-Beat-Boxing: Music Retrieval For The DJ. In *Proceedings of the 5th International Conference on Music Information Retrieval (ISMIR'04)*, pages 170–177, 2004.
- [32] R. J. Vlek, R. S. Schaefer, C. C. A. M. Gielen, J. D. R. Farquhar, and P. Desain. Shared mechanisms in perception and imagery of auditory accents. *Clinical Neurophysiology*, 122(8):1526–1532, 2011.
- [33] J. Willander and S. Baraldi. Development of a new clarity of auditory imagery scale. *Behaviour Research Methods*, 42(3):785–590, 2010.

EMOTION BASED SEGMENTATION OF MUSICAL AUDIO

Anna Aljanaki
Utrecht University
A.Aljanaki@uu.nl

Frans Wiering
Utrecht University
F.Wiering@uu.nl

Remco C. Veltkamp
Utrecht University
R.C.Veltkamp@uu.nl

ABSTRACT

The dominant approach to musical emotion variation detection tracks emotion over time continuously and usually deals with time resolutions of one second. In this paper we discuss the problems associated with this approach and propose to move to bigger time resolutions when tracking emotion over time. We argue that it is more natural from the listener's point of view to regard emotional variation in music as a progression of emotionally stable segments. In order to enable such tracking of emotion over time it is necessary to segment music at the emotional boundaries. To address this problem we conduct a formal evaluation of different segmentation methods as applied to a task of emotional boundary detection. We collect emotional boundary annotations from three annotators for 52 musical pieces from the RWC music collection that already have structural annotations from the SALAMI dataset. We investigate how well structural segmentation explains emotional segmentation and find that there is a large overlap, though about a quarter of emotional boundaries do not coincide with structural ones. We also study inter-annotator agreement on emotional segmentation. Lastly, we evaluate different unsupervised segmentation methods when applied to emotional boundary detection and find that, in terms of F-measure, the Structural Features method performs best.

1. INTRODUCTION

Improving automatic music emotion recognition (MER) methods is crucial to enhance accessibility of large music collections for both personal and commercial use. Driven by this interest, the MER field greatly expanded in the last decade. One of the fundamental MER problems is tracking emotion over time, or music emotion variation detection (MEVD). This problem is usually approached by a continuous approach to MER (dynamic MER), when the emotion of a piece of music is predicted on a second-by-second basis. Though dynamic MER does not actually assume that emotion in music should change every second, the current methods tend to work on very low time resolutions both by choosing rather short excerpts where no serious musical development could occur (e.g., 15 seconds) and by

collecting the ground truth with certain task demands on the annotators. It has been notoriously difficult to collect a ground truth for MEVD with a reasonable inter-annotator agreement, and the reason may lie in the fact that musical meaning is usually communicated during bigger time spans than several seconds, and it is therefore difficult and unnatural for the listeners to evaluate their emotional response to music in such a way. Though it might still be interesting and important to track musical change over time, the question should be raised whether change on such a short time scale is actually an expression of musical emotion or *the means of creating* emotional expression on a higher level (e.g., *accelerando* or *crescendo*).

A bordering MER field (static MER) studies identification of emotion in somewhat longer musical segments. Static MER methods usually deal with excerpts of 15 to 30 seconds. It is natural for listeners to describe musical content by applying emotional labels to musical excerpts or complete pieces. This kind of labels are used by most music services to categorize their data. However, the real world problem of MEVD requires music to be presegmented into fragments with stable emotion. This problem is usually just neglected by static MER methods, which often use ground-truth excerpts picked by randomly sampling the audio and filtering out the excerpts that receive contradictory ratings from experts. Also, sometimes the problem is solved (or rather avoided) by trying to pick the most representative part of the song for classification (e.g., chorus).

Hence, many questions about emotional segmentation of music remain unsolved. What is a typical length of an emotionally stable fragment in music? (Ironically, both static and dynamic MER methods usually deal with musical excerpts of more or less the same lengths, ranging from 15 to 45 seconds in an attempt to cover as much different music as possible while reducing the annotation burden.) Is emotional segmentation explained by structural segmentation? How many emotional boundaries are there typically in a piece of music? Which segmentation methods work best when applied to emotional boundary detection?

These are the questions that we are going to deal with in this paper. For these purposes we assemble a dataset of 52 double-annotated pieces from the RWC music database [6], which also have structural annotations in the SALAMI dataset [13]. We obtain a little under 2000 annotated emotional boundaries (around 630 from each of the annotators). We compare emotional and structural segmentation of music, analyze the inter-annotator agreement and the



© Anna Aljanaki, Frans Wiering, Remco C. Veltkamp. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Anna Aljanaki, Frans Wiering, Remco C. Veltkamp. "Emotion based segmentation of musical audio", 16th International Society for Music Information Retrieval Conference, 2015.

average stable segment length. Then we apply four segmentation algorithms to emotional segmentation problem and benchmark them on our dataset. Though the dataset is not big, a formal evaluation of emotional segmentation performance has never been conducted before.

In this work, we are not going to deal with MER in a traditional sense (predicting emotion from a musical excerpt). There already exist numerous state-of-the-art approaches to this problem [20]. Here we will address the question how to do the preprocessing step before static MER, i.e., emotional segmentation of music.

The rest of the paper is organized as follows. In section 2 we describe related research. In section 3 we explain why dynamic MER methods, at least in their current form, might not produce a good solution to the MEVD problem. In section 4 we analyze the obtained emotional segmentation. In section 5 we compare different segmentation methods when applied to a problem of detecting emotional boundaries in music. Section 6 concludes the paper.

2. RELATED WORK

Though the problem of emotional boundary detection has not yet been addressed systematically, there exist MER methods that can be applied to this problem, and we will review them in this section. For a more general overview of MER, [20] can be consulted.

2.1 Static MER for MEVD

The most simple approach to MEVD when using a static MER method is detecting emotion over time using a sliding window. This method would give a distorted result when a sliding window has an emotional boundary in it. In [21], a sliding window of ten seconds and 1/3 overlap is used to segment a music piece, and a fuzzy classifier is trained to detect the emotion of the segments. In [9] it is suggested that a homogeneous music segment is usually around 16 seconds, and therefore a sliding window of 16s is used to detect the boundaries by comparing feature distributions from neighboring windows. This approach is shown to be viable, though many questions are left open. For instance, only two features — intensity and timbre — are tested, and the evaluation is conducted only on 9 pieces. A similar approach is attempted in [15] to solve a multi-label classification problem (with two sliding windows of 10s and 30s). It is concluded that a more sophisticated emotional segmentation strategy is needed. Multi-label classification approaches recognize that one musical piece can express a variety of emotions and several labels are applicable to one piece. However, the music is often still handled in the same way as in the static MER approach. A short excerpt (e.g., 30s) is selected ([16], [17]), and several labels are applied to it, which addresses the problem of musical ambiguity, but not musical change. As opposed to this approach, in [18] a multi-label classification was applied to whole musical pieces, which were pre-segmented using aligned lyrics annotations on an assumption that most often emotion is stable within one sentence. Then, a hi-

erarchical Bayesian model was applied to a task of multi-label classification. Due to the absence of ground-truth on emotional boundaries in [18], it is left unclear how well the annotated sentences in the lyrics actually correspond to emotional structure of the musical piece.

To answer the question of what is the typical length of musical segments that represent stable emotion, Xiao et al. tried to classify excerpts of different lengths by emotion and found that excerpts of 8 or 16 seconds have a better classification accuracy than excerpts of 4 or 32 seconds [19]. This experiment gives an indirect indication of emotional segmentation resolution.

2.2 Dynamic MER

Dynamic MER methods are usually trained on time-series of annotations, typically with a resolution of 1 or 2 Hz. In Korhonen et al. [7], musical emotion is modeled as a function of musical features using system identification techniques. In [11], conditional random fields were used to model continuous emotion with a resolution of 11×11 in valence-arousal space. A similar strategy was employed in [4], where dynamic texture models were trained corresponding to quadrants of resonance-arousal-valence model and applied to predict musical emotion continuously. When separate models are trained to predict different emotions, emotional boundary detection occurs naturally. This approach might be problematic, however, due to lack of resolution in the emotional space. Also, for boundary detection it might be more important to keep track of the local context and relative changes in musical attributes rather than predict an absolute rating at every moment. This is why unsupervised methods might work very well in this case.

3. MOTIVATION

While static MER methods cannot deal with emotionally non-homogenous music, dynamic MER methods approach this problem by taking the fragmentation to the extreme (the typical resolution of a dynamic MER method is 1 second), which might create even more problems than it solves. Firstly, the output (per-second emotion prediction) produced by a dynamic MER method is not easily interpretable and useful. Secondly, it seems that musical emotion is not conceptualized in this way by listeners.

3.1 Analyzing dynamic MER ground-truth

Dynamic MER relies on human ground-truth in the form of per-second emotional annotations, which are typically recorded from an annotator continuously moving their cursor in a one or two-dimensional space [1, 14]. It seems that this task is extremely difficult for humans, which is, in particular, indicated by a very low inter-annotator agreement as compared to static annotations (where, due to task subjectivity, it is also not very high). For the MediaEval dataset [1], the average Kendalls W is 0.23 ± 0.16 for arousal and 0.28 ± 0.21 for valence, and for the Mood-Swings Lite dataset [14] the mean Kendall's W is 0.21 ± 0.14 for arousal and 0.23 ± 0.17 for valence. All these

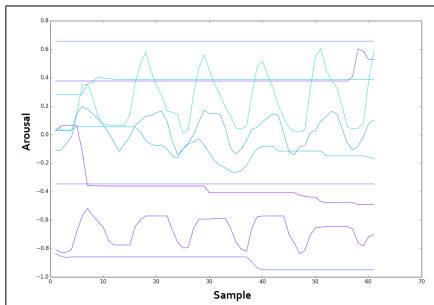


Figure 1. Dynamic annotation of 45 seconds of audio from [1]. One third of the annotators react to every beat of slow music by a peak in arousal.

numbers indicate weak agreement. There are several typical problems arising when annotating music continuously:

1. A dimensional annotation interface has an absolute scale. For instance, on an axis with a slider controlling valence, the leftmost side represents the most miserable music imaginable, and the rightmost the most ecstatic one. Giving absolute ratings is relatively easy when evaluating music statically (comparing a piece to all existing music). When comparing piece with itself over time, humans tend to think of occurring changes relatively. This leads to a huge difference in magnitude of given ratings, though the direction of change can be indicated uniformly (e.g., see Figure 1).
2. Though it is not explicitly requested from the annotators to move their cursor at all times, the task demands (short excerpt, necessity to track and respond continuously) lead to some of the annotators evaluating every single musical event (e.g., see Figure 1). This results in annotations on widely different ‘zoom level’.

We argue that continuous annotation is so difficult (albeit through training in the lab and a careful selection of complete music pieces it is possible to obtain satisfying results [3]) because it is unnatural for humans to evaluate their emotional response on a per-second basis, since emotional expression occurs on a much larger time-scale. Though through years of exposure to music listeners acquire an ability to associate certain timbres with genre and emotion, and a crude emotional interpretation is possible even from short sounds snippets of 300ms [8], we believe that real-life emotional interpretation of music is much more complex and happens during longer time spans, most certainly when it concerns induced emotion.

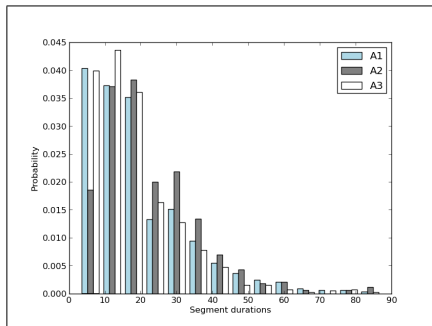


Figure 2. Histogram of segment durations for the three annotators separately.

4. ANALYSIS OF EMOTIONAL BOUNDARIES

4.1 Data

The dataset consists of 52 complete pieces [6] from Pop, Jazz and Genre (the latter contains rock, soul, world etc. music) collections of RWC music database. We picked the pieces that already had SALAMI [13] annotations in order to compare structural and emotional segmentation. The SALAMI annotations for these pieces are single-keyed, our annotations are triple-keyed in order to enable measuring agreement.

The three annotators received instructions to mark when emotion of the piece changes. There were no explicit instructions as to what could be interpreted as an emotional boundary. They were also instructed to mark the transitions between stable emotional states as separate sections, in case those were long enough to be perceived as separate ‘transition states’. In practice, this meant for instance marking long diminuendo (fade-out) at the end of a musical piece as a separate section.

In total, annotators marked 562, 602 and 746 emotional boundaries, respectively. The dataset is available from the website osf.io/jpd5z.

Evaluation metric	A2→A1	A3→A2	A1→A3
Precision @ 0.5	0.47	0.43	0.52
Recall @ 0.5	0.48	0.33	0.55
F-measure @ 0.5	0.46	0.37	0.67
Precision @ 3	0.73	0.88	0.72
Recall @ 3	0.76	0.79	0.88
F-measure @ 3	0.73	0.77	0.78

Table 1. Inter-annotator boundary retrieval with a tolerance window of 0.5 and 3 seconds.

4.2 Inter-annotator agreement

The mean number of boundaries per piece was 12.2 (median = 11.5). The average segment length was $19.5 \pm 18s$.

Figure 2 shows the histograms of segment lengths from the three annotators. We can see that the distribution is skewed, 90% of intervals are shorter than 37 seconds. Annotators 1 and 3 have annotated more short segments than annotator 2, which was caused mostly by their different decisions about short (1–3 seconds) transition segments in music (e.g., short pauses between verse and chorus).

Unfortunately, segmentation tasks are not well-adapted for formal inter-annotator agreement calculation. We perform the standard F-measure evaluation as is common in the literature [13]:

$$F_1 = 2 \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \quad (1)$$

Table 1 shows the F-measure at 0.5 and 3 seconds. The metrics are similar to those obtained for the structural segmentation task, though a bit lower for a 0.5s window [13]. It seems that 0.5s window is too strict for these particular annotations. This might be caused by the nature of the task. Though some emotional boundaries are rather abrupt, others are smeared by a transitional musical process necessary for an emotion to modulate from one state to another.

4.3 Structural segmentation explaining emotional segmentation

In order to check how well emotional segmentation is explained by structural segmentation we compared the emotional boundary annotations to structural boundaries in the SALAMI dataset. The SALAMI dataset contains hierarchical annotations on multiple levels — musical function (verse, chorus, etc.), lead instrument, and musical similarity on large and small scale. Table 2 shows the precision, recall and F-measure obtained when predicting emotional segmentation from structure. From the table we can see that about 69 to 80% of the emotional boundaries coincide with large section boundaries. More than a half of the boundaries coincide with the lead instrument change. Small-scale similarity was not included in the table because of the abundance of small-scale boundaries (meaning close to 100% recall and very low precision). We also didn't include the 0.5s time resolution, because emotional segmentation seems to be less precise than structural and 0.5s time resolution is too detailed.

It is important to note that, with regard to F-measure, the emotional annotations when retrieved from each other have a bigger score than with any of the structural segmentation annotations.

5. SEGMENTATION METHODS EVALUATION

Segmentation methods are usually categorized into homogeneity, novelty and similarity based methods. We argue that for emotional boundary detection only the first two categories are relevant, because an emotional boundary is usually signified by changes in loudness, timbral properties, harmony, instrumentation, etc., and though it might coincide with repetitive segments (i.e., chorus), there is no straightforward connection between them. Hence, in this section we are mostly going to evaluate homogeneity and novelty based methods, namely Convex NMF [10], Mood Tracking [9], the classic method by Foote [5] and Structural Features [12]. We implemented the Mood Tracking method as described in the article, and adapted an implementation¹ of the rest for our purposes (i.e., feature extraction, thresholds etc. as described below).

All of these methods are unsupervised and take as input time-series of features extracted from audio. We extract both low (mfcc, chroma, energy, dissonance and other spectral features) and high-level (scale, tempo, tonal stability) beat-synchronized audio features using Essentia [2]. Beats are determined using the Essentia BeatTracker algorithm. All the music files have 44100 Hz sampling rate and are converted to mono. To extract low-level timbral features we use a half-overlapping window of 100ms, and a window of 3 seconds for high level features. The features are smoothed with median sliding window, normalized, and resampled according to detected beats (see Figure 3a).

We use the same feature set to evaluate all the algorithms. Many segmentation algorithms limit themselves to using only MFCC or chroma features, but through experimentation with different feature sets we found that adding other spectral and high-level features significantly improves the performance on our dataset.

To combine the annotations, we decided to select only the boundaries which were marked by all the annotators with a tolerance window of 3 seconds. We will call them *common*. It can be assumed that the boundaries present in all the three annotations are the most prominent and important ones.

5.1 Summary of evaluated methods

5.1.1 Foote

Foote's method [5] relies on a self-similarity matrix (composed using pairwise sample comparisons). A short-time

¹ <https://github.com/urinieto/SegmenterMIREX2014>

Evaluation metric	Functions			Large scale			Instruments		
	A1	A2	A3	A1	A2	A3	A1	A2	A3
Precision @ 3	0.61	0.68	0.67	0.63	0.63	0.67	0.52	0.50	0.51
Recall @ 3	0.74	0.78	0.75	0.69	0.80	0.75	0.55	0.55	0.58
F-measure @ 3	0.65	0.71	0.69	0.64	0.68	0.69	0.50	0.50	0.55

Table 2. Retrieving emotional segmentation from structural segmentation

Evaluation metric	C-NMF				SF				Foote				MT (enh.)			
	C	A1	A2	A3	C	A1	A2	A3	C	A1	A2	A3	C	A1	A2	A3
P@3	.27	.35	.36	.47	.33	.43	.49	.57	.31	.38	.41	.50	.18	.28	.27	.34
R@3	.71	.67	.69	.67	.67	.61	.68	.61	.72	.67	.72	.66	.43	.47	.47	.41
F@3	.36	.43	.45	.52	.41	.47	.55	.56	.39	.45	.50	.53	.23	.34	.33	.35

Table 3. Performance of investigated methods on emotional segmentation task (F-measure).

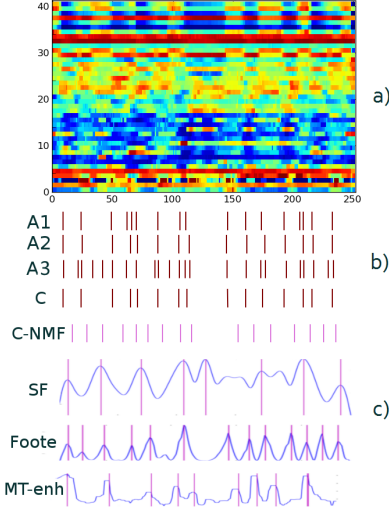


Figure 3. An illustration of the boundary detection process on the *Radezky March* by J. Strauss Sr. a) Beat-synchronized features. b) Annotations. c) Novelty curves and detected boundaries.

Gaussian checkerboard-shaped kernel is slid over the diagonal of the matrix, resulting in a novelty curve. The boundaries are detected by picking the peaks on the novelty curve. We experimented with different distance measures to compute the SSM and found that standardized euclidean distance gave the best results, which is computed between two vectors u and v as follows:

$$\sqrt{\sum (u_i - v_i)^2 / V[x_i]}, \quad (2)$$

where V is the variance vector; $V[i]$ is the variance computed over all the i th components of the points. We set the size of the checkerboard kernel to the size of the average emotionally stable segment — 20 seconds.

5.1.2 Convex NMF

The Convex non-negative matrix factorization method [10] (Convex NMF) uses a convex variant of NMF in order to

divide the audio features into meaningful clusters. This algorithm focuses both on finding segments and grouping them by similarity. If a NMF of input feature matrix X is FG, Convex NMF adds a constraint to the columns of the matrix $F(f_1, f_2, \dots, f_n)$ that the columns should become convex combinations of the features of X :

$$f_i = x_1 w_{1j} + \dots + x_p w_{pj} = X w_j, \quad j \in [1 : r], \quad (3)$$

where x_p is a column of matrix X , r is a rank of decomposition, and $w_{ij} \geq 0, \sum_j w_{ij} = 1$. This makes columns f_i interpretable as cluster centroids. We set the rank of decomposition to 2.

5.1.3 Mood Tracking

A method by Lu et al. [9] finds boundaries by comparing the audio features extracted from the two consecutive windows of 16 seconds and computing a difference between them. A novelty curve is formed using an obtained difference feature, from which peaks are picked. The difference between the consecutive windows is computed using divergence shape measure:

$$D_{i|i+1} = \frac{1}{2} \text{Tr} [(C_i - C_{i+1})(C_{i+1}^{-1} - C_i^{-1})], \quad (4)$$

where C_i and C_{i+1} are the covariance matrices of features of windows i and $i + 1$. Then, confidence of boundary is computed:

$$\text{Conf}_{i|i+1} = \exp\left(\frac{|D_{i|i+1} - D_{\text{mean}}|}{D_{\text{var}}}\right), \quad (5)$$

where D_{mean} and D_{var} are the mean and variance of all divergence shapes for this song. From a list of boundary confidences the boundaries are retrieved by satisfying conditions of being a local maximum and exceeding a local adaptive threshold.

We implemented the method as it was described in [9], but it didn't work well in its original form on our data. The constraint of 16 seconds was too conservative and adaptive threshold window was too narrow. We describe an optimized version below. The optimized version performs on average about 10% better than the original method, and we only show the performance of the optimized version in Table 3.

5.1.4 Enhanced Mood Tracking

The best results with Lu et al. method were obtained using a window of 4 seconds to compute the divergence shape measure. We smoothed the boundary confidence vector with a median filter before peak picking. To pick the peaks, we select a maximum in a neighbourhood of 10 beats in case it exceeds both of the two threshold – a moving average and half of the global average.

Though the performance of the method improved with modifications, it still performed worse than other methods in our evaluation.

5.1.5 Structural Features

The Structural Features (SF) method is both homogeneity and repetition based. It uses a variant of lag matrix to obtain structural features. The SF are differentiated to obtain structural features. The SF are differentiated to obtain a novelty curve, on which peak picking is performed. The SF method calculates self-similarity between samples i and j as follows:

$$S_{i,j} = \Theta(\varepsilon_{i,j} - \|x_i - x_j\|), \quad (6)$$

where $\Theta(z)$ is a Heaviside step function, x_i is a feature time series transformed using delay coordinates, $\|z\|$ is a Euclidean norm, and ε is a threshold, which is set adaptively for each cell of matrix S . From matrix S structural features are then obtained using a lag-matrix, and computing the difference between successive structural features yields a novelty curve.

5.2 Evaluation results

Table 3 shows the results obtained in evaluation. We only use a tolerance window of 3 seconds, because for our dataset a tolerance window of 0.5s is too strict. From the table we can see that the SF method consistently shows the best results in terms of F-measure. The method proposed in [9] consistently shows the worst results.

6. DISCUSSION

In this paper we discussed the problems associated with dynamic MER and argued that these problems originate from the unnaturally low time resolutions that dynamic MER is usually dealing with (Section 3). We proposed to move to bigger time resolutions by tracking emotionally stable segments over time and identifying transitions between them. We call this problem emotion based segmentation, and conduct a formal evaluation procedure, which has not been done before for this task.

We collected data on emotional segmentation of music; in total about 2000 emotional boundaries were annotated. In general, the annotators could agree rather well when identifying stable emotional segments, the inter-annotator F-measure was comparable to the one obtained for, supposedly less ambiguous, structural segmentation task, except for the very high resolution level (0.5 s). In terms of F-measure the emotional annotations coincide with each other better than any of the structural segmentation levels. That means that there exist some robust and important

emotional boundaries which are not explained by structural segmentation.

We compared emotional and structural segmentation and found that emotional boundaries coincide with structural boundaries very often. About half of the emotional boundaries were accompanied by a lead instrument change. Approximately 25% of the emotional boundaries did not coincide with the structural boundaries. For instance, an emotional change can occur within a structural section due to a modulation to a different tonality.

We found that the average length of stable emotional segment is approximately 20 seconds. This finding could be used to calculate a suitable length of musical excerpts to be employed for MEVD algorithms development and evaluation. Namely, we believe that length of such excerpts should be several times bigger than 20 seconds.

We evaluated different unsupervised segmentation algorithms on the task of emotional segmentation and found that the SF method performed best. This segmentation method is different from the second best Foote's method by incorporation of repetition-based criteria along with homogeneity-based ones. This shows that sequences of emotionally stable segments, probably, repeat in the same way as structural sequences, and therefore repetition-based cues are useful for emotional boundary detection. This finding goes against our initial intuition that novelty and homogeneity cues must be the only ones important to detect emotional change. The Mood Tracking method was demonstrated to be least useful. This method only uses a very narrow local context to find the discontinuities in a feature matrix, which appears to be not enough. We also found that employing higher level audio features, along with traditional chroma features and MFCCs, improves the performance of the methods on emotional segmentation task.

Though SF's method performed reasonably well, its performance was still much worse than the performance achieved by best methods for structural segmentation, which is a more mature area of research now. Developing better emotional segmentation methods is a crucial task to enable applying static MER algorithms to real world problems. We leave this task for future work, which can be facilitated by the data provided in this study.

7. ACKNOWLEDGEMENTS

We thank Kayleigh Hagen and Valeri Koort for assistance with the data annotation. This research was supported by COMMIT/.

8. REFERENCES

- [1] Anna Aljanaki, Mohammad Soleymani, and Yi-Hsuan Yang. Emotion in music task at mediaeval 2014. In *Working Notes Proceedings of the MediaEval 2014 Workshop*, 2014.
- [2] D. Bogdanov, N. Wack, E. Gomez, S. Gulati, P. Herrera, and O. Mayor. *Essentia: an audio analysis library*

- for music information retrieval. In *International Society for Music Information Retrieval Conference*, pages 493–498, 2013.
- [3] Eduardo Coutinho and Angelo Cangelosi. Musical emotions: Predicting second-by-second subjective feelings of emotion from low-level psychoacoustic features and physiological measurements. *Emotion*, 11(4):921–937, 2011.
- [4] J. Deng and C. Leung. Dynamic time warping for music retrieval using time series modeling of musical emotions. *IEEE Transactions on Affective Computing*, PP(99), 2015.
- [5] J. Foote. Automatic audio segmentation using a measure of audio novelty. In *Proceedings of the IEEE International Conference of Multimedia and Expo*, pages 452–455, 2000.
- [6] M. Goto, H. Hashiguchi, T. Nishimura, and R. Oka. Rwc music database: Popular, classical, and jazz music databases. In *Proceedings of the 3rd International Conference on Music Information Retrieval*, pages 287–288, 2002.
- [7] M.D. Korhonen, D.A. Clausi, and M.E. Jernigan. Modeling emotional content of music using system identification. *IEEE Transactions on Systems, Man, and Cybernetics*, 36(3):588–599, 2006.
- [8] C. L. Krumhansl. Plink: thin slices of music. *Music Perception: An Interdisciplinary Journal*, 27(5):337–354, 2010.
- [9] L. Lu, D. Liu, and H.J. Zhang. Automatic mood detection and tracking of music audio signals. *IEEE Transactions on Audio, Speech, and Language Processing*, 14(1):5–18, 2006.
- [10] O. Nieto and T. Jehan. Convex non-negative matrix factorization for automatic music structure identification. In *Proceedings of the 38th IEEE International Conference on Acoustics Speech and Signal Processing*, pages 236–240, 2013.
- [11] E. M. Schmidt and Y.E. Kim. Modeling musical emotion dynamics with conditional random fields. In *Proceedings of the 2011 International Society for Music Information Retrieval*, 2011.
- [12] Joan Serra, Meinard Muller, Peter Grosche, and Josep Lluís Arcos. Unsupervised music structure annotation by time series structure features and segment similarity. *IEEE Transactions on Multimedia, Special Issue on Music Data Mining*, 2014.
- [13] J. B. L. Smith, J. A. Burgoyne, I. Fujinaga, D. De Roure, and J. S. Downie. Design and creation of a large-scale database of structural annotations. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 555–560, 2011.
- [14] J. A. Speck, E.M. Schmidt, B.G. Morton, and Y.E. Kim. A comparative study of collaborative vs. traditional annotation methods. In *Proceedings of the 2011 International Society for Music Information Retrieval Conference*, 2011.
- [15] J.-H. Su, Y.-C. Tsai, and V. S. Tseng. Empirical analysis of multi-labeling algorithms for music emotion annotation. In *IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*, pages 1–6, 2013.
- [16] K. Trohidis, G. Tsoumakos, G. Kalliris, and I. Vlahavas. Multilabel classification of music into emotions. In *Proc. 9th International Conference on Music Information Retrieval*, pages 325–330, 2008.
- [17] A. Wiczorkowska, P. Synak, and Z. W. Ras. Multi-label classification of emotions in music. In *Intelligent Information Processing and Web Mining*, pages 307–315, 2006.
- [18] B. Wu, E. Zhong, A. Horner, and Q. Yang. Music emotion recognition by multi-label multi-layer multi-instance multi-view learning. In *Proceedings of the ACM International Conference on Multimedia*, pages 117–126, 2014.
- [19] Z. Xiao, E. Dellandrea, W. Dou, and L. Chen. What is the best segment duration for music mood analysis. In *In Proceedings of the IEEE International Workshop on Content-Based Multimedia Indexing*, pages 17–24, 2008.
- [20] Y.-H. Yang and H. H. Chen. Machine recognition of music emotion: A review. *ACM Transactions on Intelligent Systems and Technology*, 3(3), 2012.
- [21] Y.-H. Yang, C.-C. Liu, and H. H. Chen. Music emotion classification: A fuzzy approach. In *Proceedings of the 14th Annual ACM International Conference on Multimedia*, pages 81–84, 2006.

Oral Session 6

User & Community

MIREX GRAND CHALLENGE 2014 USER EXPERIENCE: QUALITATIVE ANALYSIS OF USER FEEDBACK

Jin Ha Lee

University of Washington
jinhalee@uw.edu

Xiao Hu

University of Hong Kong
xiaoxhu@hku.hk

Kahyun Choi, J. Stephen Downie

University of Illinois
{ckahyu2, jdownie}@illinois.edu

ABSTRACT

Evaluation has always been fundamental to the Music Information Retrieval (MIR) community, as evidenced by the popularity of the Music Information Retrieval Evaluation eXchange (MIREX). However, prior MIREX tasks have primarily focused on testing specialized MIR algorithms that sit on the back end of systems. Not until the Grand Challenge 2014 User Experience (GC14UX) task had the users' overall interaction and experience with complete systems been formally evaluated. Three systems were evaluated based on five criteria. This paper reports the results of GC14UX, with a special focus on the qualitative analysis of 99 free text responses collected from evaluators. The analysis revealed additional user opinions, not fully captured by score ratings on the given criteria, and demonstrated the challenge of evaluating a variety of systems with different user goals. We conclude with a discussion on the implications of findings and recommendations for future UX evaluation tasks, including adding new criteria: Aesthetics, Performance, and Utility.

1. INTRODUCTION

Since 2005, the Music Information Retrieval (MIR) community has benefited from the Music Information Retrieval Evaluation eXchange (MIREX), an annual evaluation event led by researchers at University of Illinois [7]. MIREX has had a significant contribution to the field as it allows system developers to test and improve their MIR algorithms. However, as the field matures, the current state of the art is increasingly deemed sufficient to support an acceptable degree of efficiency and effectiveness in various conventional MIREX tasks, resulting in the glass ceiling effect [1,2,11]. A number of researchers have also pointed out the limitations of MIREX, including the dominance of a system-centered approach and the lack of consideration for real users [11,12,14,19].

In response to the feedback received from the MIR community, the MIREX grand challenge was held in 2014. This was substantially different from any of the past evaluation tasks in two respects: 1) the focus of evaluation shifted to include the front end of the system (i.e., how users interact with the system), and 2) the submissions were complete MIR systems that can employ vari-

ous MIR techniques rather than individual algorithms. This marks a shift of the evaluation paradigm, since all the MIREX evaluation tasks have been focused on the back end, with the front end being largely ignored [11,15].

Three different MIR systems participated in the Grand Challenge 2014 User Experience (GC14UX). In this paper, we present the findings from analyzing the results of GC14UX, focusing on the free-text user responses. The goal of the paper is twofold: 1) understanding how users reacted to which aspects of the systems in their responses, and 2) using that knowledge to improve the design of future MIR UX evaluation tasks. In particular, we seek to answer the following research questions:

Q1. Which aspects of MIR systems were most important to users, as evidenced by the responses?

Q2. Based on users' responses, are there any evaluation criteria we should consider revising or adding for future iterations of MIR evaluation of user experience?

2. BACKGROUND

2.1 User-centered Evaluation in MIR

As pioneers in user-centered evaluation in MIR, Pauw et al. [17,18,20] conducted a series of user evaluation tasks to examine an interactive playlist generation system. Several user-centered measures were considered, including time on tasks, number of actions, preference, ease of use, and usefulness. Although the evaluation was confined to one specific MIR system, it is noteworthy that they considered the front-end interface and the user's interaction in the earlier days of MIR system evaluation. Hoashi et al. [9] also conducted a user evaluation of visualization interfaces for MIR systems based on subjective measures such as perceived accuracy and enjoyability.

Despite such efforts, most MIR evaluation research is still based on a system-centered approach without involving users. While this makes sense for some of the micro-level tasks, ultimately many algorithms that are being evaluated will be implemented as features in complete MIR systems. Therefore it is important to consider how users determine the usefulness and value of the systems. Hu and Kando [10] also emphasized the need for user-centered evaluation in MIR based on their finding that only a weak correlation existed between user-centered measures and system-centered measures in their evaluation experiment of MIR systems.

Leaving aside the shortage of user-centered evaluation in our field, the evaluation in the few aforementioned studies has been mostly limited to specific algorithms or



© Jin Ha Lee, Xiao Hu, Kahyun Choi, J. Stephen Downie. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Jin Ha Lee, Xiao Hu, Kahyun Choi, J. Stephen Downie. "MIREX Grand Challenge 2014 User Experience: qualitative analysis of user feedback", 16th International Society for Music Information Retrieval Conference, 2015.

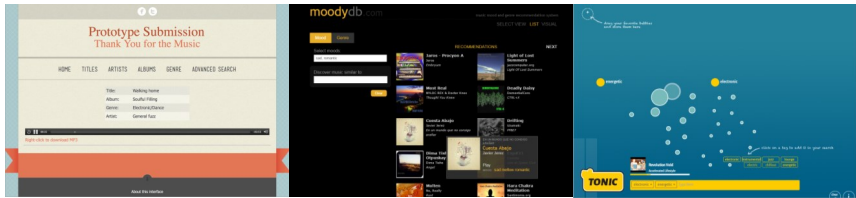


Figure 1. Screenshots of *Thank You for the Music*, *Moody*, and *Tonic*.

functions such as playlist generation algorithms and recommender systems [14]. This has been attributed to the lack of complete MIR systems ready for evaluation within the MIREX framework [11,15]. Consequently, attempts to conduct a holistic user-centered evaluation of MIR systems had to be done with existing commercial music services. For example, Lee and Price [15] examined how Nielsen’s usability heuristics [16] can be applied to evaluate multiple aspects of user experience for services like Pandora, Spotify, etc. As the MIR field is maturing, there is also a growing recognition that we are ready to evaluate complete and full-featured systems incorporating various sub-components with helpful interfaces [6,11]. Therefore, GC14UX was held, aiming to inspire the development of complete MIR systems and a holistic evaluation of user experience with those systems.

2.2 GC14UX Evaluation Framework and Process¹

The dataset used in GC14UX was a sample of 10,000 tracks with the CC-BY (Creative Commons Attribution) license from the Jamendo collection², for the purpose of avoiding any potential copyright issues. All tracks had song and album titles, artist name, and at least two genre tags. To guide the evaluators, a user task was created based on several criteria: 1) a common and realistic MIR task, 2) a task specific enough to help evaluators judge how successful the results are, 3) a task not tied to a particular MIR technique, and 4) a task that can be reasonably accomplished with the given dataset. The final task was determined as follows: “You are creating a short video about a memorable occasion that happened to you recently, and you need to find some (copyright-free) songs to use as background music.”

An online evaluation platform was set up so that evaluators could easily access the MIR systems through a web browser. The invitations were circulated through mailing lists within the MIR community. Evaluators were asked to interact with the systems and rate their scores on a seven-point Likert scale for the following criteria:

- **Overall Satisfaction:** How would you rate your overall satisfaction with the system?
- **Learnability:** How easy was it to figure out how to use the system?
- **Robustness:** How good is the system’s ability to warn you when you are about to make a mistake, allow you to recover, or retrace your steps?

- **Affordance:** How well does the system allow you to perform what you want to do?
- **Feedback:** How well does the system communicate what is going on?

Evaluators were also given an opportunity to provide their comments in an open text field.

2.3 Participating Systems and Quantitative Ratings

There were a total of three systems that participated in GC14UX: *Thank You for the Music* (hereinafter, *Thank You*), *Moody*, and *Tonic* (Figure 1)³. The design and functionality of the three MIR systems varied to some extent. *Thank You* provides users with access to a music collection through a more traditional music digital library interface, offering music search by title, album, genre, and artist. *Moody* is a recommender system in which a music collection can be browsed based on mood and genre. *Tonic* is a tag-based discovery system with a highly interactive interface utilizing pre-defined tags to find songs.

The three systems received mean scores between 4.15 and 5.37 across all criteria [11]. *Tonic* received the best score in Affordance (4.71), Feedback (4.79), and Overall Satisfaction (OS) (5.11). *Thank You* scored the highest in Learnability (5.37) with an OS of 4.15. *Moody* led in Robustness (4.53) with an OS of 4.63. However, the results of the Kruskal-Wallis test [5] showed that only the OS category had significant differences across systems [11].

3. ANALYSIS OF USER FEEDBACK

3.1 Codebook and Coding Process

We employed content analysis, a widely used qualitative data analysis method as described in [13], to uncover and code common themes in the 99 user responses. On average, there were 69 words in a response (median=51, max=259, min=2). The codebook was developed through an iterative process involving test-coding a subset of data and revising the codes for clarity. Table 1 presents detailed information on all the codes that emerged from the user responses. Each user response contained an average of 3.17 excerpts, each representing a particular code. The codes were organized into seven higher-level categories based on topical similarity. The count of excerpts for each code and the percentage calculated over the total number of excerpts (314) are also reported in the table.

¹ For more detailed information on the framework, see [11].

² <https://www.jamendo.com/en/welcome>

³ Accessible at: <http://bit.ly/1zqz1m0> (*Thank You*), <http://bit.ly/1R3fNdr> (*Moody*), <http://bit.ly/1GU7GLO> (*Tonic*)

	Categories	Codes	Definition	#	%	
Evaluation Criteria	Aesthetics	attractiveness	The user specifically talks about the visual appeal of the interface.	27	8.6	
	Affordance	access	The user specifically comments on an ability to access original music files within the system.	7	2.2	
		play function	The user specifically talks about the music play function in the system, including various aspects of the player such as the interface and features.	25	8.0	
		save function	The user specifically talks about some kind of save function like a bookmark function allowing users to revisit the page, ability to save the selected songs, or preservation of specific system settings set by the user.	12	3.8	
		search/browse	The user specifically mentions topics related to searching or browsing music based on metadata (e.g., artist name, song/album title, genre, mood labels), including advanced search, auto-complete, and finding similar items.	91	29.0	
	Feedback	clarity	The user specifically talks about the clarity of functions or labels provided.	39	12.4	
	Learnability	ease of use	The user talks about how easy, intuitive, and user-friendly it is to use the system and complete their desired task.	40	12.7	
		help	The user comments on help provided in the system such as guidelines, tutorials, or instructions.	9	2.9	
	Performance	bugs/glitches	The user specifically talks about bugs/glitches in the system that cause it to produce incorrect or unexpected results, or behave in unintended ways.	5	1.6	
		response time	The user specifically talks about the response time (i.e., the length of time taken for a system to react to a given event).	8	2.5	
		search results	The user specifically talks about the quality of search results and how they are presented to the user.	32	10.2	
	Utility	usefulness	The user talks about the overall usefulness of the system, as well as its usefulness for the given evaluation task.	13	4.1	
	Additional aspects	External factor	dataset	The user specifically notes the effects and/or limitations of using a particular dataset for the evaluation task.	6	1.9
		Sentiment	positive	The user expresses positive feelings in terms of a particular code.	107	34.1
negative			The user expresses negative feelings or desires for specific functions/features in terms of a particular code.	198	63.1	

Table 1. Summary of codebook.

The first six categories correspond to particular evaluation criteria. We can observe that three of these categories were used as evaluation criteria in GC14UX (in bold). Codes matching the criterion Robustness did not emerge from coding user responses. The External factor category contains the code dataset that was used to mark the responses noting limitations of the experience due to variables that were not controllable by system developers. We also had an “Other” code used for uncommon but relevant part of responses that did not fit into existing codes (e.g., comments on scalability issues, mobile device compatibility, etc.). Codes in the Sentiment category (i.e., positive and negative) were used in conjunction with another code to note users’ feelings regarding that code.

3.2 Inter-coder Reliability

To ensure consistent application of codes, two coders were recruited. The coders independently coded a subset of user excerpts (42% of all excerpts) and Cohen’s kappa coefficient [4] was calculated to measure their agreement. Table 2 shows that all the kappa coefficients for each code fall in the range of good (.60-.74) or excellent agreement (.75-1.0) [3,8]. The Pooled Kappa statistic summarizing the overall results across all the codes [21] was .884, suggesting an excellent agreement.

Code	Kappa value	Agreement level
save function	1.00	excellent
bugs/glitches	1.00	excellent

negative	0.98	excellent
positive	0.97	excellent
play function	0.95	excellent
response time	0.92	excellent
help	0.91	excellent
dataset	0.88	excellent
attractiveness	0.87	excellent
clarity	0.86	excellent
usefulness	0.85	excellent
ease of use	0.82	excellent
search/browse/metadata	0.80	excellent
search results	0.80	excellent
access	0.66	good

Table 2. Kappa coefficients for each code.

3.3 Tabulation of Codes

Table 3 shows the counts of positive excerpts for each system, sorted by the sum of all counts for each code. We can observe that participants liked *Thank You* for more functional reasons (e.g., search/browse, access to music files, search results) whereas they liked *Tonic* for aesthetics and usability aspects (e.g., attractiveness, ease of use, usefulness) in addition to functional reasons (e.g., play function, save function). *Moody’s* scores were fair across most of the codes except save function, access to music files, and search results. Overall, *Tonic* had the highest number of positive excerpts, with *Thank You* and *Moody* having approximately the same numbers.

	<i>Thank You</i>	<i>Moody</i>	<i>Tonic</i>	Sum
search/browse	14	10	4	28
ease of use	8	9	10	27
attractiveness	0	7	11	18
usefulness	1	1	6	8
play function	1	2	3	6
save function	0	0	6	6
access to music files	4	0	0	4
clarity	1	1	1	3
help	0	1	2	3
response time	1	1	1	3
search results	1	0	0	1
Total	31	32	44	107

Table 3. Tabulation of positive codes.

We also tallied up the counts of negative excerpts for each system (Table 4). Negative excerpts also include desires for additional features/functions, so a high count does not necessarily mean that participants disliked the system. *Moody* had the highest number of negative excerpts, mostly for search/browse, which was also the most commonly mentioned aspect across all three systems. Evaluators had strong opinions about the search function in *Moody*, also evidenced by the highest number of counts in search results. For *Tonic*, improving the clarity and help was important, in addition to play function.

	<i>Thank You</i>	<i>Moody</i>	<i>Tonic</i>	Sum
search/browse	19	34	10	63
clarity	5	10	20	35
search results	3	11	9	23
play function	3	7	9	19
ease of use	4	2	7	13
attractiveness	4	4	1	9
save function	2	4	0	6
dataset	2	3	1	6
help	0	1	5	6
bugs/glitches	2	1	2	5
response time	3	1	1	5
usefulness	2	3	0	5
access to music files	0	1	2	3
Total	49	82	67	198

Table 4. Tabulation of negative codes.

When we tabulate the counts based on the top-level categories and compare the counts for positive and negative excerpts for each category, we can observe with which aspects evaluators were most satisfied and dissatisfied (Table 5). Across all three systems, Affordance, Performance, and Feedback had more negative excerpts, suggesting these aspects need to be improved upon. Learnability, Aesthetics, and Utility had more positive excerpts overall, although notably *Thank You* had no positive excerpt for Aesthetics.

Top level	<i>Thank You</i>	<i>Moody</i>	<i>Tonic</i>	Sum
Affordance +	19	12	13	44
Affordance -	24	46	21	91

Learnability +	8	10	12	30
Learnability -	4	3	12	19
Feedback +	1	1	1	3
Feedback -	5	10	20	35
Performance +	2	1	1	4
Performance -	8	13	12	33
Aesthetics +	0	7	11	18
Aesthetics -	4	4	1	9
Utility +	1	1	6	8
Utility -	2	3	0	5

Table 5. Tabulation of codes at the top level categories.

4. DISCUSSION OF CATEGORIES AND CODES

4.1 Aesthetics

Aesthetics consists of a single code regarding the overall attractiveness of the system. While this aspect was not included in the GC14UX evaluation criteria, it may be appropriate to consider adopting it for future iterations. Most excerpts coded with attractiveness were about how appealing the visual interface was, with a few comments about the use of white space, clean interface, use of animation, and background color. The importance of this aspect is well-captured in the following response:

"What's funny is that while [Thank You] allows me to search and browse, I really liked the graphic nature of the previous two interfaces. I don't necessarily think this interface performs any less well than the others--"

4.2 Affordance

Affordance consists of four codes related to particular features or functions of the system. For access, most of the excerpts mentioned that *Thank You* was the only system where users could download the songs. To some participants, that meant that the system was "complete," and gave them "real results."

Excerpts coded with play function tended to be more negative, mentioning evaluators' desires to have more control in which part of the song they are playing. Some evaluators did appreciate that *Tonic* plays the selected songs starting in the middle (e.g., *"I like the fact that the selected pieces start playing from the middle, giving an immediate sense of the general mood and texture of the piece"*), but more evaluators wanted to be able to select from multiple options themselves:

"I would like to have an [sic] checkable option, "start playing from beginning"/"start playing from the middle" (or 25%, 30%, 40%), because sometimes [what] is important [is] the beginning, and sometimes (mostly) the mood of whole song."

Evaluators also commented negatively on the fact that they had to go through another step for playing the music in *Thank You* and *Moody* (e.g., *"I'd expected to start playing a track whenever I clicked on its cover, instead of having to wait for the pop-up and click 'play.'"*). The lack of visibility of the play button/slider was also noted for *Moody* and *Tonic* (e.g., *"The 'play'-slider is a bit small"; "difficult to find play button for the next song"*).

With regard to the save function, *Tonic* had multiple positive excerpts on the usefulness of the bookmark function, which was missing in other systems:

"there is a function at the top left corner for users to save their favorite results, it is convenient for user to compare the music later and choose the best result..." [Tonic]
"i wish there were a way for me to create[,] like a list, collection, playlist, or save or favorite multiple songs for comparison or reconsideration." [Moody]

The system remembering user settings was also important; evaluators noted that in *Thank You*, the player does not keep the selected volume level when a new song is loaded, and in *Moody*, switching between the mood and genre tab discards the selected search parameters. The save function code is somewhat related to the Robustness criterion in GC14UX; users want features that will help them trace back and return to previous results, although no excerpts were related to recovering from an error.

Overall, the search/browse code was applied most often, excluding the sentiment codes. *Thank You* had the highest number of positive excerpts due to the fact that multiple search options were provided (i.e., text search, form search, and advanced search) and users had the most control over how the search could be conducted (e.g., *"Its searching technique is very comprehensive and fully developed, which is excellent for users to carry out detailed and accurate search"*). The auto-complete features in *Moody* and *Tonic* were also appreciated by multiple evaluators. However, there was still a lot to be desired from the search/browse functions in all three systems. For *Thank You*, the lack of a browsing mechanism and inability to get recommendations were noted. Evaluators also commented on the limitation of genre categorization:

"...about 255 songs are identified as unknown, it may cause inconvenience to the users as they do not know the type of song, they must spend time to listen [to] it first."

For *Moody*, nine excerpts specifically asked for an ability to combine both mood and genre for search. Some wanted more labels for mood and genre, and others noted the lack of a free-text search option. For *Tonic*, a few evaluators commented on the inaccuracy of certain labels and a lack of vocabulary control:

"...the connection from tags to audio content does not always seem to be 'correct'...Especially if more than two tags are combined, there seem to be some problems."
"Moreover, maybe due to the vocabulary control, when I type 'cheerful,' no result is found, I have to type 'happy' instead, so the system is not flexible enough."

4.3 Feedback

This category consists of a code "clarity" that is about how intuitive and clear the functions and labels were. *Tonic* had 20 negative excerpts that were primarily about evaluators having trouble understanding what information the different design constructs are trying to convey (e.g., meaning of the histogram, size of the bubble) or what the result of a particular user action was:

"For instance, I noticed some bars on the left side, each corresponding to one of the search terms, that varied in height along with the bubble, which also resized. What is that? What does it mean when I move bubbles around?"

Similar concern was also raised for *Moody* (e.g., *"what does the size of the image mean?"*). In addition, a couple of evaluators pointed out that they had a hard time figuring out what the "discover music similar to" function was supposed to do. For *Thank You*, several evaluators commented on misunderstanding genre ID as the count of items under a particular category.

4.4 Learnability

This category contained two codes: ease of use and help. Overall, there were a lot more positive than negative excerpts regarding the ease of use across all three systems. Simple, intuitive, and user-friendly interface design was appreciated for *Moody* and *Tonic*. In general, evaluators also found the basic search interface in *Thank You* easy to use. Negative excerpts were on issues like the page layout or too much text (*Thank You*) or opinions based on a comparison with other systems (e.g., *"Tonic is not that easy to use when comparing with Moody"*).

For the help code, evaluators commented positively on the usefulness of a short introduction on how to use the system for *Tonic* but still desired more explanation on the meaning of design elements. For *Thank You* and *Moody*, clear searching guidelines and limitations (e.g., *"Maybe it should say somewhere that the similarity search only works for artists in the database"*) were desired.

4.5 Performance

Of the three codes belonging to the Performance category (i.e., bugs/glitches, response time, search results), search results was most commonly used, and primarily with negative sentiment. For *Thank You*, the ability to sort the results was appreciated but different sorting criteria were desired. The lack of a sorting mechanism was also mentioned for *Moody*. In addition, three evaluators stated that they wanted to know how many results there are for a particular search, as well as an option to switch between AND and OR connectors. For *Moody* and *Tonic*, several evaluators commented that they did not agree with or could not understand the results:

"The returned music doesn't really fit the moods, especially 'romantic.'" [Moody]
"I wrote: 'piano' and 'jazzy,' and just in the middle between these two main bubbles I found the song 'Salmacis - Arkangel,' which is not piano nor jazzy at all." [Tonic]

The evaluators' reactions to response time tended to vary, even for the same system, possibly due to varying Internet connection speeds and different levels of expectation. Bugs and glitches in scrolling, music playback, and entering data were also mentioned a few times, but they could also depend on the resolution setting or other configurations of the evaluators' machines and browsers. Therefore, it is important to note that what we are seeing

is simply users' interpretation of how well the system performed rather than the objective performance level.

4.6 Utility

"Usefulness" was the only code in this category, noting the general usefulness of the system as well as its appropriateness for the specified user task. *Tonic* had all positive excerpts as evaluators deemed that the tag-based browsing interface worked well for unknown music. The negative excerpts on *Thank You* and *Moody* mostly showed that evaluators wanted more features and functions. For *Thank You*, one evaluator noted that the search interface is limiting for the given evaluation task, which is about finding music for editing a personal video, since there are no content-based features.

4.7 External Factor

Comments on the limitation of the dataset were captured using the code dataset in this category. Six excerpts marked with this code were all negative, mostly stating that evaluators' unfamiliarity with the songs hindered their ability to effectively use the systems. This was especially true for *Thank You*, as evaluators could not issue searches using metadata such as artist name or song title. One evaluator also noted the difficulty in ascertaining the cause of unsuccessful results:

"...maybe the Jamendo collection is not very good for the task because of its variability: do we really not have good results or are systems unable to find them?"

5. IMPLICATIONS ON UX EVALUATION IN MIR

Based on the user responses and the experience of running GC14UX, we discuss three main implications for future UX evaluation tasks in MIR:

1) Adjustment of evaluation criteria

We recommend considering new criteria, Aesthetics, Performance, and Utility, in future UX evaluation tasks. The quantitative ratings showed that the difference of the scores in the "Overall Satisfaction" was statistically significant, but the differences in the other four criteria were not. This suggests that perhaps there are additional evaluation criteria affecting users' overall satisfaction. Based on the responses, the visual aesthetics of the system seem especially important; it is noteworthy that a large proportion of positive excerpts for *Tonic*, the most highly rated system, were based on "Aesthetics". "Aesthetics" might be the missing piece that can explain the differences observed in the "Overall Satisfaction". We also recommend rethinking the criterion "Robustness"; this may be difficult to evaluate given the limited time evaluators have to interact with the systems in the MIREX framework.

2) A better dataset and more user tasks

As some users pointed out, lack of familiarity with the songs in the dataset hindered their search/browse experience. In addition, a single user task for evaluation seems limiting, as MIR systems can serve a wide variety of use cases and scenarios. This was in fact the case in GC14UX

as the three evaluated systems were designed to serve different goals (e.g., *Thank You* for known-item searches, *Moody* for mood and genre-based search/browsing, and *Tonic* for exploring new music based on tags). For future UX evaluation, it might be worthwhile to consider establishing multiple user tasks, and perhaps something more common (e.g., playlist generation, recommendation) rather than trying to creating a task suitable for the dataset.

3) Focus on evaluation rather than competition

In addition to a common user task for evaluation, it may be fruitful to consider asking system developers to define a user task for which they want their system to be evaluated, as a secondary task. This makes sense considering that many commercial MIR systems are often targeted to support specific MIR tasks (e.g., Pandora for online radio function, Shazam for music identification), which was also the case for the three evaluated systems from GC14UX. We do acknowledge that this means we will not be able to directly compare the evaluation results of multiple systems. However, we strongly believe that the community should move away from considering this evaluation as a competition where ranking the systems is the primary goal. If we treat this as an opportunity to evaluate the systems in order to improve the design of all participating systems rather than being able to claim one system is better than the other, this issue will naturally dissolve. In case of GC14UX, the differences in scores for the three systems were not substantial; even for the single category where there was a statistically significant difference among the scores (i.e., Overall Satisfaction), the difference between the best- and the worst-performing systems is less than one point in a seven-point Likert scale (5.11 vs. 4.15). What would truly benefit our community as a whole is learning from the feedback about what users need and want, which will inform us on how to improve the design of MIR systems in general.

6. CONCLUSION AND FUTURE WORK

GC14UX was the very first attempt in conducting a holistic evaluation of user experience for complete MIR systems in the history of MIREX. Therefore, reflecting on our experience and deliberating on how to improve future UX evaluation is critical. Our findings indicate which aspects of the systems most concerned users, and how we can use that knowledge to improve the design of and criteria for future UX evaluation. We discussed three key implications for future UX evaluation: 1) consider three new criteria in future UX evaluation tasks, 2) seek a better dataset to improve evaluators' ability to effectively use the features and judge the quality of the results, and select more user tasks to reflect the diversity of the systems, and 3) focus on evaluation for the improvement of systems rather than competition. We hope to continue UX evaluation as a regular task within MIREX, and redesign the task with new use scenarios and datasets in the future. We also plan to widen our pool of evaluators so that we can do a comparative analysis of how MIR experts and general users evaluate their experiences.

7. REFERENCES

- [1] J.-J. Aucouturier and F. Pachet: "Improving timbre similarity: how high's the sky?" *Journal of Negative Results in Speech and Audio Sciences*, Vol. 1, No. 1, pp. 1-13, 2004.
- [2] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri: "Automatic music transcription: breaking the glass ceiling," *Proceedings of the International Society for Music Information Retrieval Conference*, pp. 379-384, 2012.
- [3] D. V. Cicchetti: "Guidelines, criteria, and rules of thumb for evaluating normal and standardized assessment instruments in psychology," *Psychological Assessment*, Vol. 6, pp. 284-290, 1994.
- [4] J. Cohen: "A coefficient of agreement for nominal scales," *Educational and Psychological Measurement*, Vol. 20, No. 1, pp. 37-46, 1960.
- [5] G. W. Corder and D. I. Foreman: *Nonparametric Statistics for Non-Statisticians*. Hoboken: John Wiley & Sons, 2009.
- [6] J. S. Downie, D. Byrd, and T. Crawford: "Ten years of ISMIR: reflections on challenges and opportunities," *Proceedings of the International Conference on Music Information Retrieval*, pp. 13-18, 2009.
- [7] J. S. Downie, X. Hu, J. H. Lee, K. Choi, S. J. Cunningham, Y. Hao, and D. Bainbridge: "Ten years of MIREX (Music Information Retrieval Evaluation eXchange): reflections, challenges and opportunities," *Proceedings of the International Society for Music Information Retrieval Conference*, pp. 657-662, 2014.
- [8] J. L. Fleiss: "Measuring nominal scale agreement among many raters," *Psychological Bulletin*, Vol. 76, No. 5, pp. 378-382, 1971.
- [9] K. Hoashi, S. Hamawaki, H. Ishizaki, Y. Takishima, and J. Katto: "Usability evaluation of visualization interfaces for content-based music retrieval systems," *Proceedings of the International Society for Music Information Retrieval Conference*, pp. 207-212, 2009.
- [10] X. Hu and N. Kando: "Evaluation of music search in casual-leisure situations," *Proceedings of the 5th Information Interaction in Context Symposium on IliX'14*, pp. 1-4, 2014.
- [11] X. Hu, J. H. Lee, D. Bainbridge, K. Choi, P. Organisciak, and J. S. Downie: "The MIREX Grand Challenge: a framework of holistic user experience evaluation in music information retrieval," *Journal of the Association for Information Science and Technology*, under review.
- [12] X. Hu and J. Liu: "Evaluation of music information retrieval: towards a user-centered approach," *Proceedings of the 4th Workshop on Human-Computer Interaction and Information Retrieval (HCIR)*, 2010.
- [13] K. H. Krippendorff: *Content analysis: an introduction to its methodology*. Thousand Oaks: Sage, 2013.
- [14] J. H. Lee and S. J. Cunningham: "Toward an understanding of the history and impact of user studies in music information retrieval," *Journal of Intelligent Information Systems*, Vol. 41, No. 3, pp. 499-521, 2013.
- [15] J. H. Lee and R. Price: "User experience with commercial music services: an empirical exploration," *Journal of the Association for Information Science and Technology*, DOI: 10.1002/asi.23433, 2015.
- [16] J. Nielsen: "Heuristic evaluation," In J. Nielsen, and R. L. Mack (Eds.), *Usability Inspection Methods*. John Wiley & Sons, New York, NY, 1994.
- [17] S. Pauws and B. Eggen: "PATS: Realization and user evaluation of an automatic playlist generator," *Proceedings of the International Society for Music Information Retrieval Conference*, pp. 222-230, 2002.
- [18] S. Pauws and S. van de Wijdeven: "User evaluation of a new interactive playlist generation concept," *Proceedings of the International Society for Music Information Retrieval Conference*, pp. 638-643, 2005.
- [19] M. Schedl, A. Flexer, and J. Urbano: "The neglected user in music information retrieval research," *Journal of Intelligent Information Systems*, Vol. 41, No. 3, pp. 523-539, 2013.
- [20] F. Vignoli and S. Pauws: "A music retrieval system based on user driven similarity and its evaluation," *Proceedings of the International Society for Music Information Retrieval Conference*, pp. 272-279, 2005.
- [21] H. De Vries, M. N. Elliott, D. E. Kanouse, and S. S. Teleki: "Using pooled kappa to summarize interrater agreement across many items," *Field Methods*, Vol. 20, pp. 272-282, 2008.

ACOUSTICBRAINZ: A COMMUNITY PLATFORM FOR GATHERING MUSIC INFORMATION OBTAINED FROM AUDIO

Alastair Porter^{†‡}, Dmitry Bogdanov[†], Robert Kaye[‡], Roman Tsukanov[‡], Xavier Serra[†]

[†]Music Technology Group, Universitat Pompeu Fabra, Barcelona, Spain

[‡]MetaBrainz Foundation

alastair.porter, dmitry.bogdanov, xavier.serra@upf.edu

rob, roman@metabrainz.org

ABSTRACT

We introduce the AcousticBrainz project, an open platform for gathering music information. At its core, AcousticBrainz is a database of music descriptors computed from audio recordings using a number of state-of-the-art Music Information Retrieval algorithms. Users run a supplied feature extractor on audio files and upload the analysis results to the AcousticBrainz server. All submissions include a MusicBrainz identifier allowing them to be linked to various sources of editorial information. The feature extractor is based on the open source Essentia audio analysis library. From the data submitted by the community, we run classifiers aimed at adding musically relevant semantic information. These classifiers can be developed by the community using tools available on the AcousticBrainz website. All data in AcousticBrainz is freely available and can be accessed through the website or API. For AcousticBrainz to be successful we need to have an active community that contributes to and uses this platform, and it is this community that will define the actual uses and applications of its data.

1. INTRODUCTION

One of the biggest bottlenecks in many Music Information Retrieval (MIR) tasks is the access to large amounts of music data, in particular to audio features extracted from commercial music recordings. Most approaches to tasks such as music classification, auto-tagging, music similarity and music recommendation, are based on using audio features obtained from well-established audio signal processing algorithms. This is a time consuming process that is beyond the possibilities of any individual researcher. It may not be possible for researchers to gather this much information, annotate it according to their needs, or compute the required features at the scale required for the task.



© Alastair Porter, Dmitry Bogdanov, Robert Kaye, Roman Tsukanov, Xavier Serra.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Alastair Porter, Dmitry Bogdanov, Robert Kaye, Roman Tsukanov, Xavier Serra. "AcousticBrainz: a community platform for gathering music information obtained from audio", 16th International Society for Music Information Retrieval Conference, 2015.

For example, existing datasets for genre classification are of insufficient size with respect to both the number of instances per class and the ability of these instances to accurately represent the entire musical genre space [4]. A list of datasets commonly used in MIR is provided in [1]. Half of them have fewer than 10,000 instances, although in recent years there have been attempts to create larger datasets. Building such datasets would allow research at the scale of the requirements of commercial applications.

In general however, the creation of datasets may be difficult for researchers due to a number of reasons:

- Gathering and sharing datasets require legal considerations with regard to the distribution of copyrighted material [7].
- Collections which are hand-created may be biased in their contents and annotations, especially if they are created by only one person, or if they are created for the evaluation of a specific task or algorithm (such as the GZTAN dataset, commonly used to evaluate audio feature-based genre classification algorithms [10]).

One project in recent years to address some of these issues is the Million Song Dataset (MSD) [1]. At the time of its release, this was the largest dataset of music descriptors in the MIR community and has gained a lot of attention for its size and breadth of music content, as well as the simplicity of accessing its data. The MSD relies on the EchoNest API¹ to compute its descriptors, a commercial product which is closed to academic inspection. Some downsides of this approach include:

- Implementation details of the algorithms used to compute the descriptors are unknown and it is impossible to review the quality of their implementation.
- The dataset is fixed in time, and does not appear to have been updated with new features, or music released since it was created.

The MSD has been further expanded with features computed using open source algorithms, on audio samples from 7digital.com [9]. As this dataset reflects the MSD, it is also fixed in time, and features do not represent the whole recording, but only the sample.

¹ <http://developer.echonest.com>

Based on these considerations, we believe that there is still space for a large dynamic dataset consisting of music features calculated with open algorithms.

2. ACOUSTICBRAINZ

We are introducing a new platform, AcousticBrainz,² to assist with the gathering of musical data from the music enthusiast and research community, and to provide researchers with large datasets of recordings to work with. All of the source code in AcousticBrainz is open,³ encouraging sharing of algorithms between contributors and providing the ability for people to improve on the work of others. All submitted and generated data is freely available under a Creative Commons CC-0 license (public domain).

The platform is split into three categories: feature extraction, data storage, and the creation of musical semantic information. A feature extractor, based on algorithms in the Essentia audio analysis library [2], can be downloaded by anyone who wishes to contribute data to the project. They run this extractor on their personal computer, giving audio files as input. The output of this extractor is a JSON file for each audio track containing descriptors (see Section 2.3.3). A submission tool provided with the extractor automatically uploads the JSON files to the AcousticBrainz server.

A database stores submissions and makes the data available via an API. AcousticBrainz only stores descriptors of audio, and never the actual audio itself. Submissions are identified by the MusicBrainz identifier (MBID) of the input audio file. These stable identifiers let us uniquely and unambiguously refer to a music recording, and can also let us obtain additional editorial information from MusicBrainz and from other services that also understand MBIDs.

To encourage experimentation with the data, the AcousticBrainz website lets anybody create, annotate, and share their own datasets consisting of recordings present in the database. A search interface lets users query for recordings based on editorial data from MusicBrainz or extracted features and add the results to the dataset. From these datasets users can build classifier models which can be used to estimate characteristics of any recording present in AcousticBrainz.

2.1 MusicBrainz

MusicBrainz⁴ is a community-maintained open encyclopedia of music information. It contains editorial metadata for many musical concepts, including Artists (individuals, groups, and other people associated with musical events), Releases, Recordings, and Works. It also contains relationships between items, and to other external databases. Data is entered manually by a large community of volunteers (editors), who also vote on changes made by other editors to ensure its quality. It is used by a number of commercial

companies.⁵ Every item in the database is uniquely identified by an MBID and many companies and organizations rely on these IDs as identifiers for music-related concepts. MBIDs can be used to retrieve data from external services which understand them (e.g., Last.fm, WikiData), and are also a part of the Music Ontology.

2.2 Current submission statistics

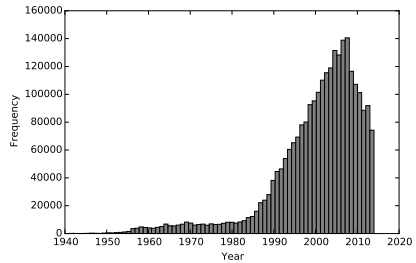


Figure 1: Release years of submissions from file metadata.

Format	Count
mp3	1,784,778
flac	777,826
vorbis	83,867
aac	64,733
alac	29,481
wmav2	4,019
other	1,320

Table 1: Number of submissions per audio codec.

At this time,⁶ the AcousticBrainz database has audio features submitted for 1,671,701 unique recording MBIDs. We keep duplicate submissions from different sources, resulting in a total of 2,747,094 submissions. For these submissions we also have metadata information available from MusicBrainz, including 99,159 artists and 165,394 releases. The duplicates consist of analyzed features of various source audio files, with differing codecs, encoders, and bit rates. These duplicates let us see real-world examples of the effect of different codecs and encoding parameters on our descriptors. We have collected submission for 538,614 unique MBIDs (807,307 including duplicates) for audio files encoded using a lossless codec (FLAC and ALAC), which is in itself a large database. The most common audio format for submitted files is MP3, with more submissions than for all other formats combined (Table 1). 94% of submitted files contain year metadata. We show a histogram of the year that submissions were released in Figure 1. The majority of tracks are from the

² <http://acousticbrainz.org>

³ <https://github.com/metabrainz/acousticbrainz-server>

⁴ <https://musicbrainz.org>

⁵ <https://metabrainz.org/customers>

⁶ July 2015

1990s and first decade of the 2000s. As tracks submitted to AcousticBrainz require a MBID this distribution may also be reflective of the content in the MusicBrainz database. The current size of the database (containing all JSON file submissions) is approximately 118GB, split between 102GB of low-level data (average file size 40kB), and 12GB of high-level (file size 4kB).

Tag	Count	Genre	%
Rock	195,837	Rock	41.15
Pop	103,486	Electronic	19.65
Classical	90,231	Pop	7.73
Jazz	88,702	Jazz	6.80
Soundtrack	79,056	Country	4.42
Electronic	71,758	Folk	3.83
Metal	44,961	Rhythm & blues	3.61
Other	42,706	Blues	2.86
Country	40,078	Hip Hop	2.23
Alternative	35,900	Classical	1.81
Alternative Rock	35,525	Asian	1.69
Folk	32,108	Caribbean	1.63
Unknown	29,413	& Latin	
Punk	27,977	Ska	0.89
Hip-Hop	24,083	Avant-Garde	0.47
Blues	23,276	Easy Listening	0.45
Indie	21,709	Comedy	0.44
Classic Rock	18,417	African	0.28
Ambient	18,074	Other	0.09
Industrial	17,816		

(a) Genre as reported in file metadata. (b) Percentages of broad genre categories.

Table 2: Genre statistics.

We find genre metadata present for 1,908,251 submissions. The top 20 genre annotations account for 52.9% of the tags used in this subset. We show the list of these genres and their counts in Table 2 (a). We also compute percentages over 691,431 recordings (41.4% of total recordings in AcousticBrainz) annotated by genre using Last.fm tags and shown in Table 2 (b). To find these broad genre labels we look up a recording by its MBID and if this fails, by the artist and track title. Last.fm tags are ranked by the most commonly applied tag. We match highly ranking tags to popular music genres found in beets, a tool for identifying, tagging, and renaming audio files,⁷. If a match occurs as a more specific subgenre, we report it as this subgenre’s parent genre. While this process is lossy (we don’t match tags which are misspelled) and subjective (not everyone agrees on genres or subgenres), we believe it nevertheless gives a good overview of the contents of the database.

2.3 Architecture

The architecture of AcousticBrainz is presented in Figure 2. The community uses the feature extractor and submission tools to send music features extracted from audio to the server. The server stores this data (which we call “low-level” data) in a database and makes it available to the rest of the community. The community can also provide classifier models (designed using the tools we provide), for inferring information from this data (which we

call “high-level” data). The high-level data is computed on the server without needing to access audio files. The community can moderate the models and the good ones are used to compute high-level data for all AcousticBrainz submissions.

2.3.1 Feature extractor and submission tool

We have created a music feature extractor using the Essentia library.⁸ We use this library for computing features because it has been successfully used in a number of similar audio analysis applications, such as Freesound, and other commercial systems. We describe the features computed by the extractor in more detail in Section 2.3.3. We distribute this extractor, written in C++, through our website⁹ as a static binary for Windows, OSX, and Linux. We use a static binary because it lets us include the same version of all of our dependencies across all platforms.

The feature extractor runs at about 20× real time, that is, a file with length 3 minutes takes 9–10 seconds to run (on an Intel i5 3.30GHz machine).

We have two clients to help the community compute features on their audio files and submit them to the AcousticBrainz server—A command-line tool written in Python, and a graphical interface written in C++ with QT. These clients automatically search for all audio files in a directory, compute their features, and send JSON files containing the features to the server using its API.

The submission tool only submits data which have been previously tagged with MBIDs. Software exists to match audio files on disk to Releases on MusicBrainz based on track lengths, file names, existing tags, and audio fingerprinting. It is possible that audio files will be tagged incorrectly, either due to user error or incorrect fingerprint matching, however we believe this to account for only a small amount of data submitted.

Each JSON file contains metadata identifying the version of the feature extractor used, including information about the exact version the source code (git commit hash) and also an increasing version number which we will change as we make incompatible changes to features in the future.

2.3.2 Server

The features of submitted tracks are stored as JSON in a PostgreSQL database. The server interface is written in Python using the Flask web application framework. An API accepts requests from clients, filters exact duplicates (where the feature extractor outputs exactly the same content for two concurrent runs on the same file), and stores the results in the database. For privacy reasons, the server stores no identifying information about submitters.

Every 30 seconds the server starts a process to search for recent submissions. For these documents the server runs a feature extractor to obtain high-level descriptors for these files (Section 2.3.4). Once the high level computa-

⁷ <https://github.com/sampsyo/beets/blob/0c7823/beetsplug/lastgenre/genres-tree.yaml>

⁸ <http://essentia.upf.edu>

⁹ <http://acousticbrainz.org/download>

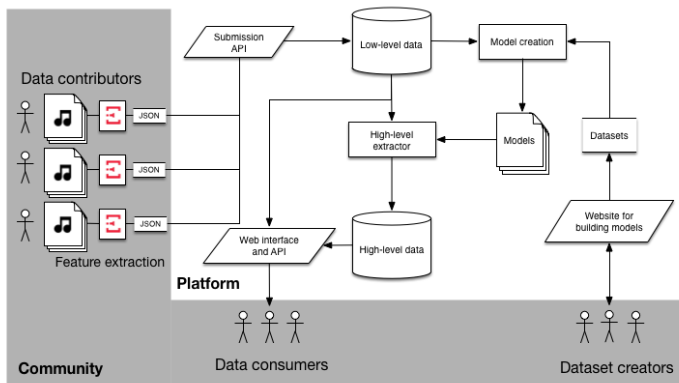


Figure 2: AcousticBrainz architecture.

tion process is complete, all of the data about the submitted track is made available to the community.

Once extracted features are made available we store all of the computed data and metadata collected from MusicBrainz in an ElasticSearch search server. This search system lets users perform queries such as finding all recordings with a particular attribute or attribute range (e.g., with a BPM between 110 and 120, or an estimated genre of jazz), or by filtering by some known metadata (such as all recordings by a particular artist).

All of the submitted and computed information is made available via the AcousticBrainz website and API. The website has a page for each submitted recording, outlining metadata, providing an overview of the low-level and high-level data, and linking to external sources, including a player to listen to the song if it is available on a public streaming service. The API gives access to the JSON documents that make up the low-level and high-level data, and access to the search interface. Documents are identified by their MBID. Groups of documents, for example all recordings in an album, can be downloaded by first getting the list of MBIDs from MusicBrainz.

2.3.3 Low-level music data

Our feature extractor computes spectral, time-domain, rhythm, and tonal descriptors. They include features characterizing overall loudness, dynamics, and spectral shape of the signal, rhythm descriptors (including beat positions and BPM value), and tonal information (including chroma features, keys and scales). All descriptors are analyzed on a signal resampled to 44.1kHz sampling rate, summed to mono and normalized using replay gain. Many of the descriptors are computed across frames and are therefore summarized by their statistical distribution (we currently do not provide per-frame information). More detailed information about the low-level data, including references to

the employed MIR and audio analysis algorithms, is provided in the official documentation for Essentia,¹⁰ or by reviewing the code.¹¹ An example of the output of the feature extractor can be seen on AcousticBrainz website.¹² We provide a list of music descriptors computed by the feature extractor and currently present in AcousticBrainz in Table 3.

2.3.4 High-level music data

Low-level data submitted by the users opens possibilities to apply data mining and machine learning techniques across the whole AcousticBrainz collection, or subsets, without needing access to audio files. In particular these techniques may allow us to infer semantic annotation of music in terms of concepts used by people when describing music (e.g., genres, styles, moods, uses of music, instrumentation, etc.). Currently, AcousticBrainz provides tools for creating datasets to represent these types of concepts and train classifier models (see Section 3). The training process is done automatically using SVM classifiers (C-SVC with polynomial or RBF kernels). A training script finds optimal data preprocessing and SVM parameterization given a ground-truth dataset of low-level data in a grid search using 5-fold cross-validation. The details on the considered parameters can be found in the classification project template in the source code.¹³ After moderation the resulting high-level data can be computed from the low-level data in the AcousticBrainz database.

Our current high-level data includes estimations done by classifiers pre-trained using a number of annotated col-

¹⁰ http://essentia.upf.edu/documentation/streaming_extractor_music.html

¹¹ <https://github.com/MTG/essentia/tree/master/src/examples>

¹² <http://acousticbrainz.org/data>

¹³ <https://github.com/MTG/gaia/tree/master/src/bindings/pygaia/scripts/classification>

low-level.*	rhythm.*	tonal.*
average loudness, dynamic complexity, silence rate 20dB / 30dB / 60dB, spectral centroid / kurtosis / spread / skewness / rolloff / decrease, hfc, spectral strongpeak, zerocrossingrate, spectral rms, spectral flux, spectral energy, spectral energyband low / middle low / middle high / high, barkbands, melbands, erbbands, mfcc, gfcc, barkbands crest / flatness db / kurtosis / skewness / spread, melbands crest / flatness db / kurtosis / skewness / spread, erbbands crest / flatness db / kurtosis / skewness / spread, dissonance, spectral entropy, pitch salience, spectral complexity, spectral contrast coeffs / valleys	beats position, beats count, bpm, bpm histogram first peak bpm / spread / weight, bpm histogram second peak bpm / spread / weight, beats loudness, beats loudness band ratio, onset rate, danceability	tuning frequency, hpcp, thpcp, hpcp entropy, key key, key scale, key strength, chords strength, chords histogram, chords changes rate, chords number rate, chords key, chords scale, tuning diatonic strength, tuning equal tempered deviation, tuning nontempered energy ratio

Table 3: Descriptors extracted by Essentia’s music extractor v1.0 currently present in AcousticBrainz. The descriptors are grouped according to the namespaces within the music extractor’s output.

Name	Source	Type	Size
genre dormund	Music Audio Benchmark Data Set [5]	Genre	1886 track excerpts, 46-490 per genre
genre rosamerica	In-house [4]	Genre	400 tracks, 50 per genre
genre tzanetakis	GTZAN Genre Collection [11]	Genre	1000 track excerpts, 100 per genre
genre electronic	In-house	Electronic music subgenres	250 track excerpts, 50 per genre
mood acoustic	In-house [8]	Sound (acoustic, non-acoustic)	321 full tracks + excerpts, 193/128 per class
mood electronic	In-house [8]	Sound (electronic, non-electronic)	332 full tracks + excerpts, 164/168 per class
timbre	In-house	Timbre colour (dark, bright)	3000 track excerpts, 1500 per class
tonal atonal	In-house	Tonality (tonal/atonal)	345 track excerpts, 200/145
danceability	In-house	Danceability	306 full tracks, 124/182 per class
ismir04 rhythm	ISMIR2004 Rhythm Classification Dataset [3]	Ballroom music dance styles	683 track excerpts, 60-110 per class
voice instrumental	In-house	Voice/instrumental music	1000 track excerpts, 500 per class
gender	In-house	Gender in vocal music (male/female)	3311 full tracks, 1508/1803 per class
mood happy	In-house [8]	Mood (happy, non-happy)	302 full tracks + excerpts, 139/163 per class
mood sad	In-house [8]	Mood (sad, non-sad)	230 full tracks + excerpts, 96/134 per class
mood aggressive	In-house [8]	Mood (aggressive, non-aggressive)	280 full tracks + excerpts, 133/147 per class
mood relaxed	In-house [8]	Mood (relaxed, non-relaxed)	446 full tracks + excerpts, 145/301 per class
mood party	In-house [8]	Mood (party, non-party)	349 full tracks + excerpts, 198/151 per class
moods mirex	MIREX Audio Mood Classification Dataset [6]	Mood (5 clusters)	269 track excerpts, 60-110 per class

Table 4: Music collections used for training high-level classifier models currently included in AcousticBrainz.

lections, some of which are commonly used in MIR (Table 4). These datasets pre-date the AcousticBrainz platform and so some of them are not yet open to inspection. We anticipate that the community can help to build better classifiers using the low-level data already submitted to AcousticBrainz.

The evaluation metrics obtained from training our current models¹⁴ show promising results. However, the accuracy and reliability of our current high-level data is under doubt, as little research on the portability of such models to the large scale has been done within MIR. We see the design of new classifier models using AcousticBrainz data as an attractive challenge for MIR researchers and we anticipate AcousticBrainz to become a platform for building classifiers on larger collections created and annotated by the community using the tools we provide (see Section 3). The high-level data within AcousticBrainz will be constantly updated using the improved classifier models proposed by the community.

3. BUILDING ANNOTATED DATASETS

We have developed an interface which lets users create datasets, comprised of a name, a list of classes, and a list

of instances for each class. These instances refer to recordings in the AcousticBrainz database, and so are referred to by MBID. MBIDs can be chosen manually, or added as the result of a search query for all recordings matching given criteria. To assist in the inspection of datasets, metadata of these recordings from MusicBrainz is also shown. Users can create these datasets individually or collaborate together to suggest classes, class boundaries, and content. We currently limit our interest to classification problems, though we see future value in allowing users to create other kinds of annotated datasets such as collections of singular types of data (e.g., music from a specific culture), user-defined lists of recordings, or sets of recordings with a freeform annotations including tags.

Once a dataset has been created, a user can choose to generate a model representing the dataset. This model is trained using the same training script used to generate our existing models (Section 2.3.4). We report to the user the accuracy of the model, giving them the chance to share the results with the wider community, or continue improving the model (Figure 3).

Once a model has been created and approved by the community we can choose to process all existing low-level data with this model in order to make these new estimations available for the community. We are able to compute high-level data at a rate of about 1000/minute using a sin-

¹⁴<http://acousticbrainz.org/data>

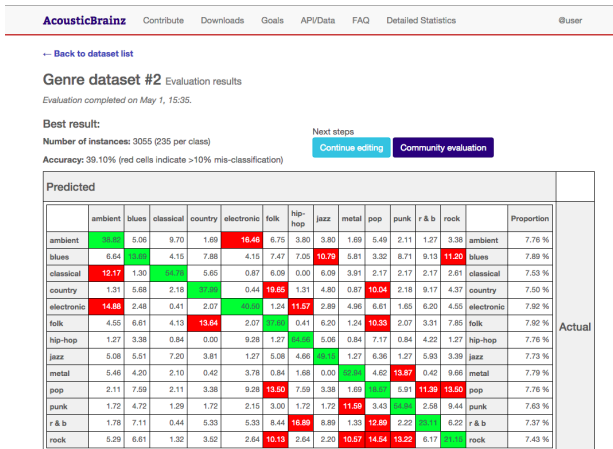


Figure 3: Results of a classification. The user can choose to continue working on improving the classifier accuracy, or submit it to the community.

gle core, and so anticipate that recomputing the dataset at its current size will only take a few days. As the dataset grows the task can be parallelized over many machines.

4. CHALLENGES AND FUTURE WORK

To keep our low-level data at the level of the state of the art in MIR, we will continue to release updates to the feature extractor, and we also encourage participation in this process. Because we rely on the good will of the community to run this extractor on their audio collections we face a trade-off between the frequency of updates and their willingness to run the extractor. We anticipate that we could release an update once or twice a year, increasing the number and quality of the features. Our high-level data will also be under constant improvement. We hope that the system that we have developed will foster collaboration to build better annotations of musically useful concepts.

Other datasets, such as the MSD contain more detailed features than those which we compute for our low-level data. The continual testing and improvement and integration of new algorithms will allow us to close this gap of feature content. Since we rely on contributions by the community, we may be missing some popular music. Continuing to solicit requests will ensure we have as broad a coverage as possible. While soliciting audio features we have to ensure that incorrect submissions are not made, either maliciously or due to incorrect metadata. We are developing a technique to determine if two submissions are identical based on their features.

Updating the feature extractor and classifier models implies compatibility problems with our data. As our submitted data includes information about the version of the ex-

tractor used to compute it, we can determine if two pieces of data computed by different versions of the feature extractor are compatible. We are compiling a dedicated audio collection to perform tests with different extractor versions and estimate the differences in feature values. These tests can also help us to assess the robustness of music features present in low-level data, the identification of which is a challenging task [12]. To take advantage of as much data as possible, we will not discard old submissions from our database when a new extractors are released. High-level data will be updated with respect to low-level data when possible. Improvements to the collection creation interface on the AcousticBrainz website will let us build datasets to use with other machine learning techniques.

We expect that the data provided by AcousticBrainz will be useful to both the MIR community and others interested in this type of data. In exchange we need the AcousticBrainz community to help in expanding the dataset and improving its quality. The interest in our platform became apparent directly after its launch when we were able to obtain features for 500,000 files in less than 3 weeks, building up to over 2.7 million submissions. The continued support of providing features and collaborating on data collection projects will ensure the success of this project.

5. ACKNOWLEDGEMENTS

This research has been partially funded by the CompMusic (ERC 267583) and SIGMUS (TIN2012-36650) projects. The authors would like to thank the entire MusicBrainz community and the members of the AcousticBrainz community who helped to test the feature extractor and contribute data to the project.

6. REFERENCES

- [1] T. Bertin-Mahieux, D. PW Ellis, B. Whitman, and P. Lamere. The million song dataset. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 591–6, 2011.
- [2] D. Bogdanov, N. Wack, E. Gómez, S. Gulati, P. Herrera, O. Mayor, G. Roma, J. Salamon, J.R. Zapata, and X. Serra. Essentia: An audio analysis library for music information retrieval. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 493–498, 2013.
- [3] P. Cano, E. Gómez, F. Gouyon, P. Herrera, M. Koppenberger, B. Ong, X. Serra, S. Streich, and N. Wack. ISMIR 2004 audio description contest. Technical report, 2006. Available online: <http://mtg.upf.edu/node/461>.
- [4] E. Guaus. *Audio content processing for automatic music genre classification: descriptors, databases, and classifiers*. PhD thesis, Universitat Pompeu Fabra, 2009.
- [5] H. Homburg, I. Mierswa, B. Möller, K. Morik, and M. Wurst. A benchmark dataset for audio classification and clustering. In *Proceedings of the International Conference on Music Information Retrieval*, pages 528–31, 2005.
- [6] X. Hu and J. S. Downie. Exploring mood metadata: Relationships with genre, artist and usage metadata. In *Proceedings of the International Conference on Music Information Retrieval*, pages 67–72. Citeseer, 2007.
- [7] D. Karydi, I. Karydis, and I. Deliyannis. Legal issues in using musical content from iTunes and YouTube for music information retrieval. In *International Conference on Information Law*, 2012.
- [8] C. Laurier, O. Meyers, J. Serrà, M. Blech, and P. Herrera. Music Mood Annotator Design and Integration. In *International Workshop on Content-Based Multimedia Indexing*, 2009.
- [9] A. Schindler, R. Mayer, and A. Rauber. Facilitating comprehensive benchmarking experiments on the million song dataset. In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 469–74, 2012.
- [10] B. L. Sturm. An analysis of the gtzan music genre dataset. In *Proceedings of the second international ACM workshop on Music information retrieval with user-centered and multimodal strategies*, pages 7–12. ACM, 2012.
- [11] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE transactions on Speech and Audio Processing*, 10(5):293–302, 2002.
- [12] J. Urbano, D. Bogdanov, P. Herrera, E. Gómez, and X. Serra. What is the effect of audio quality on the robustness of mfccs and chroma features? In *Proceedings of the International Society for Music Information Retrieval Conference*, pages 573–8, 2014.

HOW MUSIC ALTERS DECISION MAKING - IMPACT OF MUSIC STIMULI ON EMOTIONAL CLASSIFICATION

Elad Liebman

Computer Science Department
The University of Texas at Austin
eladlieb@cs.utexas.edu

Peter Stone

Computer Science Department
The University of Texas at Austin
pstone@cs.utexas.edu

Corey N. White

Department of Psychology
Syracuse University
cnwhite@syr.edu

ABSTRACT

Numerous studies have demonstrated that mood can affect emotional processing. The goal of this study was to explore which components of the decision process are affected when exposed to music; we do so within the context of a stochastic sequential model of simple decisions, the drift-diffusion model (DDM). In our experiment, participants decided whether words were emotionally positive or negative while listening to music that was chosen to induce positive or negative mood. The behavioral results show that the music manipulation was effective, as participants were biased to label words positive in the positive music condition. The DDM shows that this bias was driven by a change in the starting point of evidence accumulation, which indicates an a priori response bias. In contrast, there was no evidence that music affected how participants evaluated the emotional content of the stimuli. To better understand the correspondence between auditory features and decision-making, we proceeded to study how individual aspects of music affect response patterns. Our results have implications for future studies of the connection between music and mood.

1. INTRODUCTION

There is robust evidence that one's mood can affect how one processes emotional information. This phenomenon is often referred to as mood-congruent processing or bias, reflecting the finding that positive mood induces a relative preference for positive emotional content (and vice versa). The goal of the present study was to use a popular model of simple decisions, the drift-diffusion model (DDM; [9]), to explore how music-induced mood affects the different components of the decision process that could drive mood-congruent bias. The model, described below, can differentiate two types of bias: a) Bias due to an a priori preference for one response over the other; and b) Bias due to a shift in how the stimuli are evaluated for decision making. This

class of models has been successfully employed to differentiate these biases in perceptual and memory tasks, but to our knowledge has never been used to investigate effects of music on emotional classification. We consider the following to be our key contributions: a) We provide meaningful evidence that decision making is indeed affected by music stimuli, and analyze the observed effects; b) we study evidence of how specific auditory features are correlated with aspects of decision making.

Studies that induce mood, either through listening to happy/sad music or having participants write passages or see pictures based on a particular emotion, have shown mood-congruent bias across a range of tasks. Behen et al. [4] showed participants happy and sad faces while they listened to positively- or negatively valenced music and underwent fMRI. Participants rated the happy faces as more happy while listening to positive music, and the fMRI results showed that activation of the superior temporal gyrus was greater when the face and music were congruent with each other. In a study of mood and recall, De l'Etoile [3] found that participants could recall significantly more words when mood was induced (through music) at both encoding and retrieval. Similarly, Kuhbandner and Pekrun [6] had participants study emotional words that were printed in either black, red, green, or blue, with the hypothesis that congruent words (e.g., negative words in red, positive words in green) would show enhanced memory at test. Their findings supported the hypothesis, as memory was better for negative words shown in red and positive words shown in green.

Previous work at the intersection of musicology and cognitive science has also studied the connection between music and emotion. As Krumhansel points out [5], emotion is a fundamental part of music understanding and experience, underlying the process of building tension and expectations. There is neurophysiological evidence of music being strongly linked to brain regions linked with emotion and reward [1], and different musical patterns have been shown to have meaningful associations to emotional affectations [8]. Similarly, studies have indicated that mood also affects the perception of music [12]. Not only is emotion a core part of music cognitive processing, it can also have a resounding impact on people's mental state, and aid in recovery, as shown for instance by Zumbansen et al. [15] in the case of people suffering from Brocas aphasia. People regularly use music to alter their moods, and evidence



© Elad Liebman, Peter Stone, Corey N. White.

Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Elad Liebman, Peter Stone, Corey N. White. "How Music Alters Decision Making - Impact of Music Stimuli on Emotional Classification", 16th International Society for Music Information Retrieval Conference, 2015.

has been presented that music can alter the strength of emotional negativity bias [2]. All this evidence indicates a deep and profound two-way connection between music and emotional perception.

The structure of the paper is as follows. In Section 2 we outline the characteristics of the drift-diffusion model, which we use in this study. In Section 3 we discuss our experimental design and how data was collected from participants. In Section 4 we present and analyze the results of our behavioral study. In Section 5 we further analyze how individual auditory components correlate with the behavioral patterns observed in our human study. In Section 6 we recap our results and discuss them in a broader context.

2. THE DRIFT-DIFFUSION MODEL

This study employs the DDM of simple decisions to relate observed decision behavior to the underlying decision components. The DDM, shown in Figure 1, belongs to a broader class of evidence accumulation models that posit simple decisions involve the gradual sequential accumulation of noisy evidence until a criterial level is reached. In the model, the decision process starts between the two boundaries that correspond to the response alternatives. Evidence is accumulated over time to drive the process toward one of the boundaries. Once a boundary is reached, it signals a commitment to that response. The time taken to reach the boundary denotes the decision time, and the overall response time is given by the decision time plus the time required for processes outside the decision process like encoding and motor execution. The model includes a parameter for this nondecision time (T_{er}), to account for the duration of these processes.

The primary components of the decision process in the DDM are the boundary separation, the starting point, and the drift rate. Boundary separation provides an index of responses caution or speed/accuracy settings; wide boundaries indicate a cautious response style where more evidence needs to be accumulated before the choice is made. The need for more evidence makes the decision process slower, but also more accurate as it is less likely to hit the wrong boundary by mistake. The starting point of the diffusion process (z), indicates whether there is a response bias. If z is closer to the top boundary, it means less evidence is required to reach that boundary, so “positive” responses will be faster and more probable than “negative” responses. Finally, the drift rate (v) provides an index of the evidence from the stimulus that drives the accumulation process. Positive values indicate evidence for the top boundary, and negative values for the bottom boundary. Further, the absolute value of the drift rate indexes the strength of the stimulus evidence, with larger values indicating strong evidence and leading to fast and accurate responses.

In the framework of the DDM, there are two mechanisms that can drive behavioral bias. Changes in the starting point (z) reflect a response expectancy bias, whereby there is a preference for one response even before the stimulus is shown [7, 14]. Experimentally, response expectancy

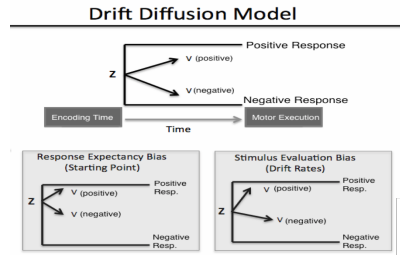


Figure 1. An Illustration of the Drift-Diffusion Model.

bias is observed when participants have an expectation that one response is more likely to be correct and/or rewarded than the other. In contrast, changes in the drift rate (v) reflect a stimulus evaluation bias, whereby there is a shift in how the stimulus is evaluated to extract the decision evidence. Experimentally, stimulus evaluation bias is observed when there is a shift in the stimulus strength and/or the criterion value used to classify the stimuli. Thus responses expectancy bias, reflected by the starting point in the DDM, indicates a shift in how much evidence is required for one response relative to the other, whereas stimulus evaluation bias, reflected by a shift in the drift rates in the DDM, indicates a shift in what evidence is extracted by the stimulus under consideration. Importantly, both mechanisms can produce behavioral bias (faster and more probable responses for one choice), but they differentially affect the distribution of response times. In brief, response expectancy bias only affects fast responses, whereas stimulus evaluation bias affects both fast and slow responses (see [14]). It is this differential effect on the RT distributions that allow the DDM to be fitted to behavioral data to estimate which of the two components, starting point or drift rates, is driving the bias observed in the RTs and choice probabilities. The DDM has been shown to successfully differentiate these two bias mechanisms from behavioral data in both perceptual and recognition memory tasks [14].

This study used the DDM approach described above to investigate how music-induced mood affects the different decision components when classifying emotional information. Participants listened to happy or sad music while deciding if words were emotionally positive or negative. The DDM was then fitted to each participant’s behavioral data to determine whether the mood induction affected response expectancy bias, stimulus evaluation bias, or both.

3. METHODS

Participants were shown words on the computer screen and asked to classify them as emotionally positive or negative while listening to music. The words were emotionally positive, negative, or neutral. After a fixation cue was shown for 500 ms, each word was presented in the center of the

screen and remained on screen until a response was given. If no response was given after 3 seconds, the trial ended as a “no response” trial. Responses were indicated with the “z” and “/” keys, and mapping between the key and response was counterbalanced across participants. The task consisted of 4 blocks of 60 trials with 20 stimuli from each word condition (positive, negative, neutral). A different song was played during each block, alternating from positive to negative music across blocks. The order of the songs was counterbalanced across subjects. The entire experiment lasted less than 30 minutes. To ensure that the results were not specific to the particular choice of songs, the entire experiment was replicated with a new set of music.

The stimuli consisted of emotionally positive (e.g., success, happy), negative (e.g., worried, sad), and neutral words (e.g., planet, sipped) taken from a previous study [13]. There were 96 words for each stimulus condition, which were matched for word frequency and letter length. From each wordpool, 80 items were randomly chosen for each participant to use in the task. Words were randomly assigned to appear in the positive or negative music blocks with the constraint that 20 of each word type appeared in every block of trials.

Publicly available music was surveyed to isolate two clear types - music that is characterized by slow tempo, minor keys and somber tones, typical to traditionally “sad” music, and music that has upbeat tempo, major scales and colorful tones, which are traditionally considered to be typical to “happy” music. Our principal concern in selecting the musical stimuli, rather than their semantic categorization as either happy or sad, was to curate two separate “pools” of music sequences that were broadly characterized by a similar temperament (described above), and show they produced consistent response patterns.

To ensure that the selected music was effective for inducing the appropriate mood, a separate set of participants rated each piece of music on a 7-point Likert scale, with 1 indicating negative mood and 7 indicating positive mood. There were 21 participants that rated the songs for Experiment 1, and 19 participants for Experiment 2. This mood assessment was done outside of the main experiment to eliminate the possibility that the rating procedure would influence the participants’ classification behavior in the primary task. The ratings showed that the music choices were appropriate. The positive songs in Experiment 1 led to more positive ratings than the negative songs. Similar results were found for the songs in Experiment two, with higher ratings for the positive songs than the negative songs. The differences between the positive and negative song ratings were highly significant for both experiments (p-values < .001 using a paired t-test, with $t(20) > 7.3$). The means and standard deviations of the scores for the songs in the two experiments are presented in table 1.

The DDM was fitted to each participant’s data, separately for positive and negative music blocks, to estimate the values of the decision components. The data entered into the fitting routine were the choice probabilities and response time (RT) distributions (summarized by the .1,

song	—Experiment 1—		—Experiment 2—	
	average	SD	average	SD
happy 1	5.14	1.24	5.15	1.29
happy 2	5.00	1.22	5.42	1.17
sad 1	2.24	1.00	2.26	1.24
sad 2	2.33	0.97	2.11	0.99

Table 1. Aggregated Likert scale results for the 8 songs used in the two experiments.

.3, .5, .7, and .9 quantiles) for each response option and stimulus condition. The parameters of the DDM were adjusted in the fitting routine to minimize the χ^2 value, which is based on the misfit between the model predictions and the observed data (see [10]). For each participant’s data set, the model estimated a value of boundary separation, nondecision time, starting point, and a separate drift rate for each stimulus condition. Because of the relatively low number of observations used in the fitting routine, the variability parameters of the full DDM were not estimated (see [9]). This resulted in two sets of DDM parameters for each participant, one for the positive music blocks and one for the negative music blocks.

4. RESULTS

The RTs and choice probabilities in Figure 2 show that the mood-induction successfully affected emotional bias. The left column shows the response probabilities, and the right column shows an RT-based measure of bias, which is taken as the median RT for negative responses minus the median RT for positive responses for each condition. Thus RT values above 0 indicate faster positive than negative responses for that condition, and vice-versa. In Experiment 1 (top row), happy music led to more “positive” responses overall. This difference was significant for neutral words and positive words, but not for negative words. For RTs, positive responses were generally faster than negative responses in the happy compared to sad music conditions, though the difference was only significant for positive words. The results from Experiment 2 largely mirror those from Experiment 1. Participants were more likely to respond “positive” in the happy music condition. This difference was significant for the negative and neutral words, but not the positive words (though there is a trend in that direction). Likewise, positive responses were relatively faster than negative responses in the happy compared to sad music conditions, though the difference was only significant for neutral and positive words.

Overall, the behavioral data show that the mood induction was effective in influencing participants’ emotional classification: positive responses were more likely and faster in the happy compared to sad music condition. These behavioral data are next decomposed with the DDM.

Figure 3 shows the DDM parameters for each experiment. Although the two bias-related measures (starting point and drift rates) are of primary interest, all of the DDM parameters were compared across music conditions.

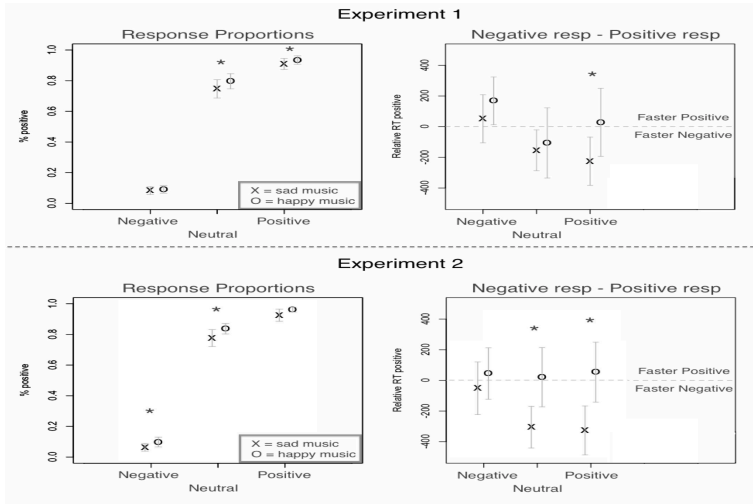


Figure 2. Response patterns for the two experiments. 1st column shows proportions of classification for the three word types. 2nd column shows normalized response time difference between positive and negative classifications for the three word types. X marks the sad music condition, O marks the happy music condition.

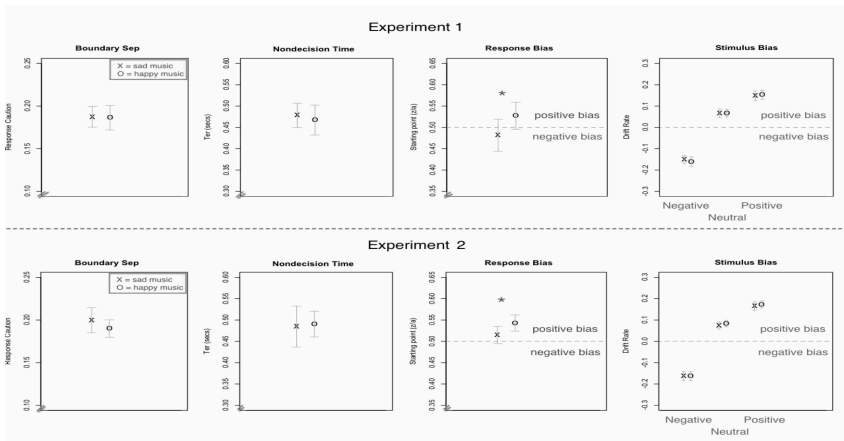


Figure 3. DDM fitted parameters - boundary separation, nonddecision time, response bias and stimulus bias. X marks the sad music condition, O marks the happy music condition. Response bias indicates a statistically significant difference between the sad and happy music conditions.

It is possible that the different music conditions could affect response caution and nondecision time. For example, the slower tempo of the sad songs could lead participants to become more cautious and have slower motor execution time. Thus all parameters were investigated. As the left columns of Figure 3 shows, the music conditions did not differentially affect response caution or encoding/motor time, as neither boundary separation nor nondecision time differed between happy and sad music blocks. Of primary interest were the starting point and drift rate parameters, which provide indices of response expectancy and stimulus evaluation bias, respectively. For starting point, there was a significant shift in response bias for both experiments, with participants favoring the “positive” response more heavily in the happy compared to sad music. This indicates that the music induced an a priori bias for one response over the other. In contrast, the music conditions had no reliable effect on the drift rates for positive, negative, or neutral words. Thus music did not influence the stimulus evaluation of the items. The DDM results show that the music-based manipulation of mood had a targeted effect on the starting point measure, which reflects an a priori response expectancy bias. There were no effects of music on response caution, nondecision time, or drift rates (stimulus evaluation bias). Thus the results show that the mood-congruent bias was driven by a change in participants’ expectancy about the appropriate response, rather than a change in how the emotional content of the words was evaluated.

5. CORRELATING RESPONSES AND MUSICAL FEATURES

The partition between “positive” and “negative” mood-inducing songs is intuitively understandable, and is indeed sufficient in order to observe the effects discussed in the previous section. This partition, however, is still somewhat arbitrary. It is of interest then to identify, on a more fundamental auditory level, how specific aspects of music affect response patterns. To this end, we considered the 8 musical segments used in this experiment, extracted key auditory features which we assume are relevant to the mood partitioning, and examined how they correlate with the participant responses we observed.

5.1 Extracting Raw Auditory Features

We focused on three major auditory features: a) overall tempo; b) overall “major” vs. “minor” harmonic character; c) average amplitude. Features (a) and (c) were computed using the Librosa library [11]. To compute feature (b), we implemented the following procedure. For each snippet of 20 beats an overall spectrum was computed and individual pitches were extracted. Then, for that snippet, according to the amplitude intensity of each extracted pitch, we identify whether the dominant harmonic was major or minor. The major/minor score was defined to be the proportion of major snippets out of the overall song sequence. We can easily confirm that these three features were indeed asso-

ciated with our identification as “positive” vs. “negative”. Having labeled “positive” and “negative” as 1 and 0 respectively, we observed a Pearson correlation of $0.7 - 0.8$ with p -values ≤ 0.05 between these features and the label. Significance was further confirmed when we applied an unpaired t-test for each feature for positive vs. negative songs (p -values $< .05$, $|t(3)| > 3$).

5.2 Processing Participant Responses

For each observed song we first aggregated all relevant subject responses. We focused on three measurements - time delay for classifying positive words as positive, time delay for classifying negative words as negative, and likelihood of classifying neutral words as positive. Time delays were normalized to a z-score per user. This alternative perspective helps verify the robustness of the effects observed in the previous section. Following this analysis step, we proceeded to fit the DDM parameter decomposition as we did in sections 3 and 4, but rather than for each song condition (“sad”/“happy”), to each song separately.

5.3 Observed Correlations

In this section we consider the effects observed when analyzing response patterns with respect to each of the three auditory features discussed in the previous subsections. Only statistically significant correlations are reported, though it’s worth noting that with a relatively small sample size in terms of songs, potentially meaningful effects might be missed due to outliers.

5.3.1 Correlation with Response Times and Bias

When we consider how the three auditory features correspond with the normalized delays when classifying positive or negative words as such, we see an interesting pattern. For all three features, there was a statistically significant negative correlation (p -value ≤ 0.05) between the average normalized response time and the feature values. Intuitively speaking, the faster the song was, the louder it was, or the more it was major in mode overall, the faster people classified positive words as positive (see Figures 4a-4c). However, no such clear correlation was observed for negative songs. This observation supports our key finding when using the drift-diffusion model, that participants were biased to label words positive in the positive music condition. When we analyzed the likelihood of associating neutral words as positive with respect to each auditory feature, the only effect that is borderline significant (p -value ≤ 0.1) is the correspondence between major mode dominance and the likelihood of associating a neutral word as positive (the more major-mode oriented the song is, the more likely people are to associate neutral words as positive) - see Figure 4d.

5.3.2 Correlation with DDM Decomposition

We analyzed the correlation between the extracted auditory features and the DDM parameters fitted for each song separately: nondecision time, response caution, response bias, and stimulus evidence (drift rate) for each word type.

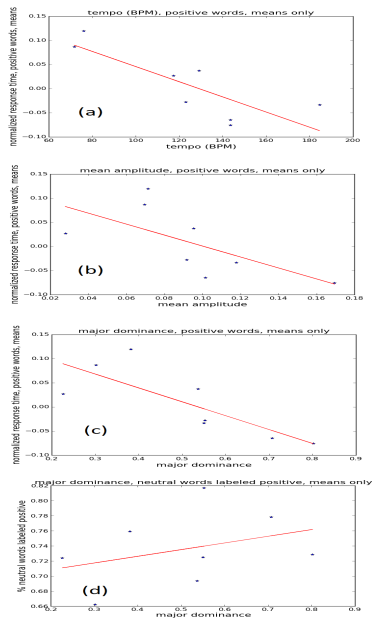


Figure 4. Scatter plots reflecting the correlation between musical features and response patterns: (a) average tempo (BPM) vs. the normalized average delay in classifying positive words; (b) average amplitude vs. normalized average delay in classifying positive words; (c) % of major-mode harmonies vs. normalized average delay in classifying positive words. (d) % of major-mode harmonies vs. the likelihood of associating neutral words as positive.

We found a statistically significant correlation ($r = 0.7 - 0.8, p < 0.05$) between the major dominance feature and the bias and positive drift rate parameters (see Figures 5a, 5b). A borderline correlation ($r = 0.62, p < 0.1$) was observed between major dominance and the neutral drift rate. These findings support the previous observations in the paper. Interestingly, we've also observed a borderline significant negative correlation ($r = -0.67; p < 0.1$) between mean amplitude and response caution, implying people are less cautious the louder the music gets (see Figure 5c).

6. DISCUSSION

There is great interest in understanding how music affects emotional processing. This study advances our understanding of this relationship through the use of the drift-diffusion model, which was used to decompose the behavioral data into meaningful psychological constructs. Participants classified words as emotionally positive or nega-

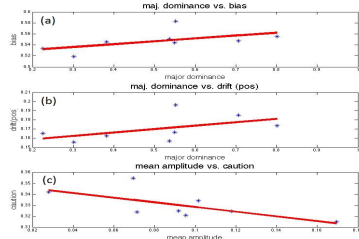


Figure 5. (a) Scatter plot of the correlation between the percentage of major-mode harmonies (major dominance) in a song and the bias component of the DDM. (b) Scatter plot of the correlation between the percentage of major-mode harmonies (major dominance) in a song and the stimulus evidence component (drift rate) for positive words in the DDM. (c) Scatter plot of the correlation between the average amplitude of a song and the response caution component of the DDM.

tive while listening to music that induced a happy or sad mood. The behavioral data showed small, but reliable effects of mood congruent emotional bias based on the music conditions. The DDM analysis of those data showed that music-induced mood had a targeted effect on the decision components, affecting response expectancy bias but not stimulus evaluation bias, response caution, or encoding/motor time. Further analysis of how specific musical traits correspond with response patterns confirmed these findings and led to interesting additional observations.

These results suggest that music-induced mood does not significantly affect how participants evaluate the emotional content of the stimuli, but rather it affects how they favor one response option independent of the actual stimulus under consideration. In other words, a negative word is just as negative while listening to sad compared to happy music, even though the classification behavior differs. Thus the mood-congruent bias appears to be driven more by the selection of the response, rather than the emotional processing of the stimulus. The distinction between these two processes is only identifiable through the DDM analysis, as it can capitalize on the RT distributions to dissociate the two decision components.

Acknowledgments

This work has taken place in the Brain and Behavior Lab, The Department of Psychology, Syracuse University, and in the Learning Agents Research Group (LARG) at the Artificial Intelligence Laboratory, The University of Texas at Austin. LARG research is supported in part by grants from the National Science Foundation (CNS-1330072, CNS-1305287), ONR (21C184-01), AFRL (FA8750-14-1-0070), AFOSR (FA9550-14-1-0087), and Yujin Robot.

7. REFERENCES

- [1] Anne J Blood and Robert J Zatorre. Intensely pleasurable responses to music correlate with activity in brain regions implicated in reward and emotion. *Proceedings of the National Academy of Sciences*, 98(20):11818–11823, 2001.
- [2] Jie Chen, Jiajin Yuan, He Huang, Changming Chen, and Hong Li. Music-induced mood modulates the strength of emotional negativity bias: An erp study. *Neuroscience Letters*, 445(2):135–139, 2008.
- [3] Shannon K de l’Etoile. The effectiveness of music therapy in group psychotherapy for adults with mental illness. *The Arts in Psychotherapy*, 29(2):69–78, 2002.
- [4] Jeong-Won Jeong, Vaibhav A Diwadkar, Carla D Chugani, Piti Sinsoonsud, Otto Muzik, Michael E Behen, Harry T Chugani, and Diane C Chugani. Congruence of happy and sad emotion in music and faces modifies cortical audiovisual activation. *NeuroImage*, 54(4):2973–2982, 2011.
- [5] Carol L Krumhansl. Music: A link between cognition and emotion. *Current Directions in Psychological Science*, 11(2):45–50, 2002.
- [6] Christof Kuhbandner and Reinhard Pekrun. Joint effects of emotion and color on memory. *Emotion*, 13(3):375, 2013.
- [7] Martijn J Mulder, Eric-Jan Wagenmakers, Roger Ratcliff, Wouter Boekel, and Birte U Forstmann. Bias in the brain: a diffusion model analysis of prior probability and potential payoff. *The Journal of Neuroscience*, 32(7):2335–2343, 2012.
- [8] Sébastien Paquette, Isabelle Peretz, and Pascal Belin. The musical emotional bursts: a validated set of musical affect bursts to investigate auditory affective processing. *Frontiers in psychology*, 4, 2013.
- [9] Roger Ratcliff and Gail McKoon. The diffusion decision model: theory and data for two-choice decision tasks. *Neural computation*, 20(4):873–922, 2008.
- [10] Roger Ratcliff and Francis Tuerlinckx. Estimating parameters of the diffusion model: Approaches to dealing with contaminant reaction times and parameter variability. *Psychonomic bulletin & review*, 9(3):438–481, 2002.
- [11] Brian McFee ; Matt McVicar ; Colin Raffel ; Dawen Liang ; Douglas Repetto. *Librosa*. <https://github.com/bmcfee/librosa>, 2014.
- [12] Jonna K Vuoskoski and Tuomas Eerola. The role of mood and personality in the perception of emotions represented by music. *Cortex*, 47(9):1099–1106, 2011.
- [13] Corey N White, Aycan Kapucu, Davide Bruno, Caren M Rotello, and Roger Ratcliff. Memory bias for negative emotional words in recognition memory is driven by effects of category membership. *Cognition & emotion*, 28(5):867–880, 2014.
- [14] Corey N White and Russell A Poldrack. Decomposing bias in different types of simple decisions. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 40(2):385, 2014.
- [15] Anna Zumbansen, Isabelle Peretz, and Sylvie Hébert. The combination of rhythm and pitch can account for the beneficial effect of melodic intonation therapy on connected speech improvements in broca’s aphasia. *Frontiers in human neuroscience*, 8, 2014.

PUT THE CONCERT ATTENDEE IN THE SPOTLIGHT. A USER-CENTERED DESIGN AND DEVELOPMENT APPROACH FOR CLASSICAL CONCERT APPLICATIONS

Mark S. Melenhorst

Delft University of Technology
Multimedia Computing Group
m.s.melenhorst@tudelft.nl

Cynthia C. S. Liem

Delft University of Technology
Multimedia Computing Group
c.c.s.liem@tudelft.nl

ABSTRACT

As the importance of real-life use cases in the music information retrieval (MIR) field is increasing, so does the importance of understanding user needs. The development of innovative real-life applications that draw on MIR technology requires a user-centered design and development approach that assesses user needs and aligns them with technological and academic ambitions in the MIR domain. In this paper we present such an approach, and apply it to the development of technological applications to enrich classical symphonic concerts. A user-driven approach is particularly important in this area, as orchestras need to innovate the concert experience to meet the needs and expectations of younger generations without alienating the current audience. We illustrate this approach with the results of five focus groups for three audience segments, which allow us to formulate informed user requirements for classical concert applications.

1. INTRODUCTION

While the Music Information Retrieval (MIR) field historically has mostly been algorithmically oriented, in recent years the community increasingly gained interest in the use and consequences of MIR technology for real-life applications rooted in user needs. Cases for a 'mentality shift' into this direction have been made in [22], [4], [20], [6], [15], and the ISMIR community includes a limited amount of active work on real-world user requirements (e.g. [3], [10], [12], [13]). However, it still seems hard to connect real-world user needs and requirements to concrete technological system and algorithmic advances [14]. When the needs and characteristics of the users are left unaddressed in technological applications, the end user remains an abstract entity, which becomes manifest in the absence of a requirements analysis and untargeted participant recruitment for formative or summative evaluations of systems involving MIR technology.

In this paper, we focus on technological application opportunities targeted at (Western) classical symphonic concert attendance. Orchestras are increasingly worried

about audience sustainability. A Flemish study confirmed the common belief that concert attendees are typically highly educated and over the age of 45 [19]. Concerns about an aging audience motivate orchestras to find creative ways to involve new audiences [11], not only with new attractive concert formats, but also with technological innovations that allow users interested in classical music to become engaged in an easy way. Examples include online concert broadcasting (e.g. Digital Concert Hall¹), smartphone-supported live program notes (e.g. LiveNote²), and enriched tablet e-magazines with second screen content (e.g. RCO Editions³). As argued in [9], MIR technology has the potential of enriching the customer experience for the users of these applications. Once users become more engaged, they are more likely to buy concert tickets, which ultimately would lead to a more diverse classical concert audience.

However, there is a trade-off between the need for innovations that attract new audiences and the risk of avoiding the alienation of the traditional audience. Since technology is not naturally associated with the classical concert experience and the allegedly conservative audience might be reluctant towards the use of technology in and around the concert hall, the importance of user acceptance cannot be underestimated. Therefore, an innovation approach is needed that combines a technology push from the MIR community with a strong technology pull from user audiences. User-centered design is an important pillar of this approach, addressing user needs from existing and new audiences, and evaluating solutions with end-users in every stage of the design process.

In this paper, we therefore demonstrate how a user-centered design approach can be used to identify opportunities for the use of (MIR) technology in classical concert applications that are grounded in the needs of different audience segments. More specifically, our study seeks to answer the following research questions:

1. What are the motivators and obstacles for different audience segments to (not) attend classical concerts?
2. How can the needs of the audience segments be translated into opportunities for the enrichment of the classical concert experience by means of technology?

Consistent with [9], we argue that the classical concert experience not only involves the concert itself, but also



© Mark S. Melenhorst, Cynthia C. S. Liem.
Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Melenhorst, M.S. & Liem, C.C.S. "Put the Concert Attendee in the Spotlight. A User-Centered Design and Development Approach for Classical Concert Applications", 16th International Society for Music Information Retrieval Conference, 2015.

¹ <https://www.digitalconcerthall.com/en/home>

² <https://www.philorch.org/introducing-livenote%E2%84%A2-nights/>

³ www.concertgebouworkest.nl/en/rco-editions

the preparation beforehand, and reflection and re-experience afterwards. The envisioned applications are intended to appeal to new audiences by yielding a stronger hedonic response on four sources of stimulation (emotions, senses, imagination, and intellect) [18], for both current and new audiences.

After discussing related work on the needs of different classical music audience segments, we outline the user-centered design approach that was taken. Subsequently, we present the results with respect to the first steps in our approach: the user requirements elicitation process that was preceded by the construction of user stories [16]. User requirements are derived from focus groups that address motivators and obstacles for classical concert attendance and that collect feedback on a set of user stories, containing ideas for the use of (MIR) technology to enrich the classical concert experience before, during and after the concert.

2. AUDIENCE SEGMENTS

While the classical concert audience sometimes perceives itself as homogeneous, in fact this is not the case [17]. To develop applications that support the needs of the classical concert audience, it is therefore important to distinguish different audience segments. Roose [19] suggests a tripartite audience segmentation. First, *passers-by* are incidental – typically younger – visitors that are not motivated by the concert performance itself, but rather by extrinsic motivations such as an evening out with friends. *Participants* comprise the core of the audience, consisting of well-informed, well-interested people that generally are not formally trained in music. In contrast, the *inner circle* consists of audience members that are professionally involved in the arts who frequently attend concerts and form a peer group for the performers. A large-scale survey conducted by [19] demonstrated that the average age for all participants was between almost 55 and 57. The educational level for all segments is higher than for the general population (above bachelor level or higher). Inner circle members are better educated than participants, who in turn are better educated than passers-by. This tripartite segmentation is used as the basis for the audience segmentation that will be used in this paper as the basis for application development.

3. MOTIVATORS AND OBSTACLES

The development of concert experience enrichment applications requires a solid understanding of why people enjoy classical concerts (*motivators*) and what *obstacles* they experience towards concert attendance. This section discusses existing literature on these, with focus on North-American and European audiences.¹

In a Flemish study, Roose [19] distinguishes between extrinsic and intrinsic motivations for concert attendance

and five classes of aesthetic dispositions. Even though the definitions of and relationships between motivations and dispositions are not precisely defined, they can shed light on why classical concerts appeal to different audiences.

Intrinsic and extrinsic motivators. Intrinsic motivations include the performers (e.g. a soloist or an orchestra), the programming, or a concert being part of a seasonal ticket. Extrinsic motivators are social motivators (advice from others, invitation from others, or spending time with friends), or attention in the media. Radbourne et al. [18] further elaborate on the social part of the experience, referred to as ‘collective engagement’. They argue that this an important determinant of the audience experience. Collective engagement can take three forms: between the audience and the performers, among audience members, and between attendees and non-attendees. Social interactions stimulate discussion about the music [18], which would facilitate learning. This in turn would improve the audience experience.

Aesthetic dispositions. [19] distinguishes five aesthetic dispositions that influence one’s inclination to attend classical concerts: emotional, escapist (e.g. change of setting to escape everyday concerns), familiarity (e.g. music one is familiar with), normative (e.g. criticize society), and innovative (e.g. experiments with the tonal system, complex rhythmic patterns, etc., with the purpose of encouraging the listener to discover new meanings in the music). The innovative disposition primarily is particularly present in well-educated, experienced audiences.

In comparison to motivators for classical concert attendance, relatively little is known about the obstacles preventing people from attending classical concerts. [11] and [4] invited participants to attend a classical concert for the first time. Responses of first-time classical concert attendees can shed light on the preconceptions with which they enter the concert hall, and the difficulties they face. From these studies three classes of obstacles can be derived: limited sense of belonging, knowledge about classical music, and richness of the experience.

Limited sense of belonging. Classical concert novices might feel overwhelmed when they enter a concert hall for the first time due to the social conventions, the etiquette, and the social interactions that occur. [4] and [11] have shown that first-time attendees have trouble with adjusting to these. [4] reported a lack of a sense of belonging as a result of age differences and differences in clothing. The limited sense of belonging because of social distance and the unknown social conventions is amplified by a limited understanding of the music. Additionally, [11] found that the lack of interaction between audience members and between the audience and performers negatively impacted the experience of first-time concert attendees, corroborating the importance of collective engagement that was suggested by [18].

Knowledge about classical music. Respondents in [4] and [11] articulated the importance of acquiring a certain level of knowledge to enjoy the concerts more. Knowledge about classical music is also related to emotions. While the emotional response is an important determinant of the audience experience [18], these emotions

¹ To the best of our knowledge, no cross-cultural comparisons involving audiences with other cultural backgrounds have been made; however, also in this paper, we will focus on Western audience.

are more likely to be evoked when the attendee has a certain level of knowledge. Currently available information sources prove to be ill-adjusted to non-regular audiences, imposing an obstacle to the learning process. Respondents in [4] complained about the program notes, which were ill-adjusted to first-time attendees in terms of vocabulary and required general background knowledge.

Richness of the experience. Classical concerts are rather different for first-time attendees compared to popular music concerts. Kolb [11] indicated that their respondents were able to pay attention during about 10 minutes per piece. They also felt that there were little opportunities for interaction between the audience and the performers, while the setting did not allow for interaction between audience members. Participants in [11] indicated that the lack of visual stimuli on the stage caused the time to go slow. Participants in both [11] and [4] reported a lack of visual stimuli, caused by both the stage set up, and the way the performers dress (referred to as ‘funeral attire’).

To the best of our knowledge, no prior academic studies exist which comprehensively address both motivators and obstacles on classical concert attendance for multiple audience segments. In the following sections, we will describe how we investigated this, with the ultimate goal of developing innovative classical concert applications.

4. DESIGN APPROACH

The development of applications that are well-aligned with the needs and preferences of the users requires a multi-stakeholder approach. *Orchestras* characterize their target audiences through marketing research. New applications need to be aligned with their business model and their marketing strategy. *Existing and new audience members* need to provide input on their needs and expectations. Throughout the development cycle, they provide feedback on prototypes of increasing fidelity. *Technology providers* (businesses and research institutes) develop the actual applications, based on academic or business ambitions, balancing technology-push with technology-pull.

In **Figure 1**, a high-level user-centered design and development process for classical concert applications is displayed, involving the aforementioned stakeholders.

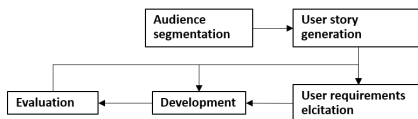


Figure 1 User-centered development process

Audience segmentation. In the work described in this paper, applications need to be tested for all parts of the concert experience: before, during, and after a concert. Based on (unpublished) marketing research from a Dutch orchestra, three audience segments were targeted: outsiders, casual consumers, and heavy consumers. In comparison to [19], *outsiders* (OS) are comparable to the passers-by or the ‘culturally-aware non-attenders’. The *casual consumers* (CC) are in between the participants and the pass-

ers-by. While they have serious interest in attending classical music, compared to participants, their concert attendance frequency is lower, as is their average age, musical knowledge level, and less natural engagement with classical music. *Heavy consumers* (HC) comprise both the inner circle and the participants.

User stories. User stories describe specific functionalities, written from the perspective of an end-user. They function as data collection probes [1] – artefacts “containing open-ended, provocative and oblique tasks to support early participant engagement responses with the design process” (p. 1077). In our work, eight user stories were constructed that each describe a set of features for smartphone or tablet applications, expected to enrich the concert experience before, during and after event attendance. The user stories – described in [16] – address the needs of all three relevant target audiences (OS, CC, HC), while at the same time, they build on opportunities from the technological and MIR domain.

The insights gained from feedback on the user stories shape the *user requirements* in a way we will describe in the remainder of this paper. In their turn, the requirements form the basis for the iterative development and evaluation of these apps.

5. USER REQUIREMENTS ELICITATION METHODOLOGY

5.1 General approach

For user requirements elicitation, five focus groups were held: two in the Netherlands (one for HC consumers, one for CC consumers) and three in Austria (targeting HC, CC and OS consumers, respectively). After signing informed consent forms a project introduction was given. Participants then introduced themselves, focusing on their music preferences. Subsequently, motivators and obstacles for attending classical concerts were discussed, introduced by the questions “What makes a classical concert such a unique experience for you?” and “What prevents you from going to classical concerts more often?”, respectively. Afterwards, participants received a booklet with the user stories, which the participants read, annotated on sticky notes, and discussed. The focus group was concluded with a questionnaire, addressing technology use, music and concert behavior, and demographics.

5.2 Participants

In the Netherlands, participants were recruited via a mailing of the Royal Dutch Concertgebouw Orchestra’s customer association, whose members fitted our CC and HC criteria. In Austria, a recruitment e-mail was sent to all students of a university. A sign-up form with questions about classical music involvement was used to divide participants over the three audience segments. **Table 1** reports participant characteristics for all focus groups.

5.3 Data analysis

After transcription, the data were analyzed using thematic analysis, a form of pattern recognition within the data, where emerging themes become analysis categories [8]. Data were analyzed with the purpose of identifying moti-

vators and obstacles for concert attendance. Feedback on the user stories was analyzed with the purpose of deriving opportunities for applications to enrich the classical concert experience. Note that even though the differences between the Netherlands and Austria are of interest to the goal of our study, other differences between the samples (e.g. age, occupational status, income, experience) prevent us from doing a valid cross-cultural comparison.

Measure	The Netherlands	
	CC	HC
N	6	13
Age	27.7 (.8)	54.7 (15.2)
Concert attendance		
> once/month	1	5
once/month		
once/quarter	4	8
once/year	1	

Measure	Austria		
	OS	CC	HC
N	7	10	4
Age	29.4 (8.2)	27.8 (11.3)	27.5 (3.8)
Concert attendance			
> once/month		1	
once a month			1
once/quarter	3	3	2
once every year	4	6	1

Table 1. Focus group participant characteristics

6. MOTIVATORS AND OBSTACLES

In this section, we present the results of devising general motivators and obstacles from the user requirements elicitation process. Transcription, analysis, and coding of the results has led to the definition of 17 motivators and 16 obstacles, of which we will discuss the most important ones, backed with statements from the discussions. Statement quotes use the following abbreviations: #n=participant ID; OS=outsiders, CC=casual consumers, HC=heavy consumers; NL=the Netherlands, AT=Austria. Statements from sticky notes do not have a participant ID, as they were collected all at once on a flip-over sheet.

6.1 Intrinsic motivators

Concert experience and musical quality. Across target groups, participants appreciate the uniqueness of the concert as a one-time event during which high-quality music is played. Participants clearly see the added value of a live concert in comparison to a recording. They felt that this was not only applicable to classical music, but also to concerts in other genres (CC-NL, OS/CC/HC-AT).

The discussions revealed that in classical concerts, attendees are motivated by the interaction between the conductor and the orchestra, between the audience and the performers, and by the orchestra members themselves. Tension and suspense fascinated the participants: “You can see tension with musicians, feeling is transmitted through the way they look and move. You can also see this from the conductor. (...) you can feel the emotion, not just audio. You don’t get this in a recording.” (#9-

CC-AT). From the OS-AT group it became apparent that this fascination not only applies to classical concerts, but also to other genres.

Escapism. For casual consumers and heavy consumers in both AT and NL, escapism – an aesthetic disposition mentioned by [19] – is an important motivator for classical concert attendance. Participants indicated that submerging themselves in an environment in which they cannot do anything else but focus on the music allows them to disconnect from their daily concerns (“At a classical concert I forget all my problems, I am not stressed, #6-CC-AT). In that sense, a classical concert was compared to a church service: “A moment to be quiet” (#2-CC-NL). Another participant emphasized the difference to listening to classical music at home: “It’s an obligation to listen to a concert in peace and quiet. I don’t succeed in doing that at home” (#5-CC-NL).

Need for cognition. People differ in the extent to which they desire to engage in cognitively effortful activities [2]. In the CC-NL and HC-NL groups, opportunities for cognitive engagement and learning motivated several participants to attend classical concerts. Curiosity about the musicians, the piece, and the performers was expressed (referred to as ‘hunger for information’; #5-CC-NL). However, this need for cognition and learning was not expressed by participants in the outsider group.

One participant in the CC-AT group connected the escapist motivator and the resulting focus on the music to an increased level of processing: “You start thinking about things. You discover new pieces”. Another participant noticed a difference in attitude with respect to learning: “Awareness and qualitative enjoyment of a piece is more important than entering the hall snobbishly, pretending that you know everything” (#5-CC-NL).

6.2 Extrinsic motivators

Social influences. Participants reported that having peers or family members with the same interests, helps to get motivated for classical concerts. One participant commented: “I notice that it works well when you know a couple of people in the orchestra. It makes things more personal. And lowers the barrier to join in” (#6-CC-NL). Furthermore, in particular the younger CC-NL group preferred a concert experience to encompass more than just the performance itself (e.g. appreciating “A drink at a bar with young people afterwards”; #5-CC-NL).

6.3 Intrinsic obstacles

Importance of classical music. The discussions revealed substantial differences between audience segments concerning the role classical music plays in people’s lives. Consistent with findings from [19] and [11], we found that participants are not exclusively focused on classical music, but are ‘culturally mobile’ [7]. One participant explained: “You just don’t visit 10 classical concerts. There is more than classical music. It’s interesting if something comes up. And that’s what our generation likes” (#3-CC-NL). Participants also mentioned that their interest in classical concerts is mood-dependent.

Preparation and risk. Substantial differences were found with respect to the effort audience segments were

willing to invest in concert preparations. While the CC-NL group requested easily consumable information, the HC-NL group was motivated to invest more time (“I can spend hours on YouTube watching videos about what a singer has done before”; HC-NL), and considered preparation as part of the pre-concert anticipation. The discussion in the OS-AT group revealed that the risk of buying an expensive ticket can be too high (“It’s expensive for just something you don’t know”, #4-HC-AT). To reduce this risk, participants felt they needed to invest time in finding information about the performers and the piece. This factor was more important in AT than NL, probably because due to the AT participants being students.

Concert setting and conventions. Consistent with [11] participants in the younger groups (OS/CC-AT; CC-NL) felt disconnected from other concert attendees, primarily as a function of the age difference “What stops me? That there are very, very, very many seniors in the hall. Sometimes that disturbs me” (#2/3/5, CC, NL). Participants also mentioned the complexity of concert conventions for novices (“a classical concert can be intimidating. Unknown. They don’t know the rules”, #4+6-CC-NL).

Richness of the experience. Results suggested that the perceived richness of the experience was dependent on both the age group and the level of engagement with classical music. Younger groups (CC-NL, OS-AT, HC-AT) noted that “The experience is richer in other genres, for classical it’s more about the music itself” (#3-HC-AT). One participant (#3) in the OS-AT group commented on the lack of surprises, knowing already what the playlist is. Interestingly, the HC-NL group considered the surprise element to be a motivator (“At every performance you become surprised by something...you hear things you won’t hear elsewhere”, #10-HC-NL). Outsiders (OS-AT) and casual consumers (CC-NL, CC-AT) commented on the lack of opportunities for physical expression. “I miss standing up. Being engrossed in music you also experience physically.” (#2-CC-NL). This radically differs from their experience with non-classical concerts.

6.4 Extrinsic obstacles

Social influences. Younger participants – most present in the OS and CC groups – indicated that their peers were less interested in classical music, causing a lack of company. This prevents the respondents from going more often, both in Austria and in the Netherlands. (#3-CC-NL, “You have to know people that also like classical music”; #9-CC-AT, “It’s easier to find friends who want to join me to a rock concert”). Other extrinsic obstacles included ticket costs, and the long time attendees needed to plan ahead when they want to attend a concert.

7. APPLICATION AND MIR OPPORTUNITIES

In this section, we aggregate user story feedback under several clustered themes. We discuss relevant feedback per theme, formulate exemplary requirements for technology-supported concert applications, and discuss integration opportunities for MIR technology.

7.1 Support with preparation

The discussion on motivators and obstacles has highlighted the importance of concert preparation across expertise levels. User stories facilitating concert preparation were well-received. The CC groups appreciated the convenience of having information in one place (“We are part of a generation that is used to large amounts of information, but also to get it presented in an easy way”; #2-CC-NL). Both the HC-NL and the CC-NL group appreciated the added value of the information, particularly historic context, for preparation before the concert, but also for better understanding during and after the concert.

When working towards concrete applications, this leads to the requirement that the applications should *offer information about the composer, the musicians, the piece, and its historical context*. MIR technology can support this by developing cross-modal and cross-performance synchronization methods, and techniques for analyzing and combining hybrid music information resources.

7.2 Need for support to understand the music.

While participants wanted to avoid overemphasizing cognitive aspects, across groups a need was expressed for understanding the music, learning about what parts one should pay attention to, and discovering unexpected new elements. Participants recognized the difficulty for novice listeners to understand and then enjoy the music “because music is hard to grasp/decipher” (CC-NL). They expressed interest in the structure of the music, the composer’s intention, the conductor’s interpretation, and the discovery of style differences in comparison to recordings. User stories that provide this support were assessed positively, in terms of their educational potential and the potential to lower the barriers for outsiders to start attending classical concerts.

In terms of application requirements, two main requirements can be extracted: the applications should *offer representations of the musical structure and the user’s attention should be attracted to parts of the music which wouldn’t have been noticed otherwise*. These interests confirm the relevance of MIR work on automated music description, performance analysis, and visualization.

7.3 Audience expansion by sharing relevant moments

The user stories included application features allowing users to annotate particularly interesting moments, to review the notes and related audiovisual content after the concert, and to share notes and their corresponding fragments through social media. Participants felt that the sharing of small fragments could function as an “opener” for people unacquainted with this type of music” (HC-NL). By sharing the experience, users can motivate their friends to attend a classical concert (“if you share this, you can tag someone along”; CC-NL).

While the opportunity to review and share particularly interesting moments *after* the concert was generally evaluated positively, taking notes *during* a concert was perceived as distracting. Participants were concerned with the impact on the concert experience (“It’s not a lecture”; HC-NL). They felt that the cognitive effort of taking notes “destroys magic of non-repeatable live experience”.

A one-button marker was frequently mentioned as a light-weight solution: (“Annotations for a specific moments, ok, but not with text, only with a marker. Which means: I want to hear this again”; CC-NL).

This leads to several application requirements: applications should *enable the user to set a marker at a particular moment during the concert by pressing a single button and enable the user to listen to marked fragments after a concert*. While the concepts of marking, annotating and sharing are somewhat related to work on social media and autotagging in the MIR community, many open questions are raised regarding temporal aspects of ‘interesting moments’, and especially the type of information to be displayed and marked.

7.4 Personalization and control

The results revealed substantial differences between and within audience segments, concerning their expectations of the concert experience, attitude towards technology, and level of classical music experience. Considering these differences, participants expressed interest in personalized information. Here, ‘personalization’ had two meanings: first, participants preferred to only receive information that is relevant to them, notwithstanding their need for a certain level of surprise in the information offered. Second, they wanted to switch on and off different layers of information to personalize their user experience.

These notions lead to two corresponding requirements formulations for applications: *the user must be able to receive personalized content by filling out a limited number of questions and the user must have control over the layers of information that are displayed*. Regarding the first point, an explicit questionnaire is suggested, as this provides both most transparency to a user, and avoids data sparsity issues. Still, it is useful to assess the potential of automated MIR profiling and recommendation techniques, in terms of usefulness and feasibility.

7.5 Caveat: interference with the concert experience.

One important caveat was brought up in every focus group: applications should refrain from interfering with the live concert experience. Participants wanted to enjoy the music without engaging in cognitive activities. This is in line with the escapism disposition from [19], which also emerged from the focus groups as a motivator. In terms of [18], an overemphasis on cognitive stimulation potentially prevents sensory or emotional stimuli from contributing to the concert experience. (“How can you combine a tablet with emotions?” HC-NL).

When tablets were discussed as a possible medium in concert halls, participants were worried about distraction by messages about everyday affairs (“You might receive a work-related e-mail that makes you tense up”, CC-NL). Second, tablets might also distract other audience members due to the light emittance of tablets in an otherwise (semi)-dark concert hall. The strongest rejection of these ideas came from the participants in the HC-NL group who wanted to keep the concert experience as it is.

This leads to a very clear and strong requirement that applications *must not distract the user or other concertgoers while listening to a live concert*. In terms of MIR

technology, this poses open challenges with respect to user experience design of in-concert applications.

8. CONCLUSIONS

In this paper we discussed a user-centered design approach to identify opportunities for technological enrichment of the classical concert experience. Departing from a tripartite audience segmentation and common motivators and obstacles for concert attendance from literature, five focus groups were conducted in which these motivators and obstacles were further refined and connected to application and MIR technology inclusion opportunities.

A trade-off was found between offering cognitive support to users and allowing users to enjoy the concert without disturbance. Light emittance, required attention by the user, and the impact on other concertgoers are the most important concerns that were voiced by the participants. In contrast, stronger support was found for ideas that improve the understanding of the music. Participants also supported ideas to relieve marked interesting moments of the concert, although the marking effort during the concert should be limited to pressing a single button.

The reported results support our plea for a detailed assessment of end-user needs and user characteristics. Our results reveal differences between individual participants with respect to their aesthetic dispositions [19], cultural mobility [7], and also the type of stimulation participants expect from a concert [18]. Furthermore, consistent with [21], the results suggest that age affects user acceptance of technology in the concert hall – with older participant being more reluctant towards changes of the concert experience. In sum, the results emphasize that what is a motivator for one attendee, can be an obstacle for another.

Classical concert applications for such a heterogeneous audience require a personalized user experience, with many opportunities to integrate advances from the MIR research agenda. At the same time, the success of resulting applications will depend on their connection to end-user needs and expectations. The chosen *presentation and contextualization of information* is a critical factor in this, which is not yet thoroughly examined with true end-user involvement in MIR.

Follow-up steps in our approach are to iteratively design and evaluate application wireframes for prototypical applications, while simultaneously developing the backend (MIR) technology. Results of consecutive evaluations will then refine and extend the requirements and opportunities presented in this paper.

9. ACKNOWLEDGMENT

This study is part of the PHENICX project, which is funded by the European Union Seventh Framework Programme FP7 / 2007- 2013 under grant agreement no. 601166. Additionally, the authors want to express their gratitude towards Marcel van Tilburg (Royal Concertgebouw Orchestra) and Marko Tkalčić (Johannes Kepler University) who have offered substantial support with the practical organization of the focus groups and the recruitment of participants.

10. REFERENCES

- [1] Boehner, K., Vertesi, J., Sengers, P., and Dourish, P.: 'How HCI Interprets the Probes'. Proc. CHI, San Jose, CA, April 28-May 3, 2007
- [2] Cacioppo, J.T., and Petty, R.E.: 'The need for cognition', *Journal of Personality and Social Psychology*, 1982, 42, (1), pp. 116-131
- [3] Cunningham, S.J., Downie, J.S., and Bainbridge, D.: "'The pain, the pain": modelling music information behavior and the songs we hate', in: 6th International Conference on Music Information Retrieval (ISMIR 2005). London, UK; 2005, pp. 474-477.
- [4] Dobson, M.C., and Pitts, S.E.: 'Classical Cult or Learning Community? Exploring New Audience Members' Social and Musical Responses to First-time Concert Attendance', *Ethnomusicology Forum*, 2011, 20, (3), pp. 353-383
- [5] Downie, J.S., Byrd, D., and Crawford, T.: 'Ten Years of ISMIR: Reflections on Challenges and Opportunities', in: 10th International Society for Music Information Retrieval Conference (ISMIR 2009). Kobe, Japan; 2009, pp. 13-18.
- [6] Downie, J.S., Hu, X., Lee, J.H., Choi, K., Cunningham, S.J., and Hao, Y.: 'Ten Years of MIREX: Reflections, Challenges and Opportunities', in: 15th International Society for Music Information Retrieval Conference (ISMIR 2014). Taipei, Taiwan; 2014, pp. 657-662.
- [7] Emmison, M.: 'Social Class and Cultural Mobility: Reconfiguring the Cultural Omnivore Thesis', *Journal of Sociology*, 2003, 39, (3), pp. 211-230
- [8] Fereday, J., and Muir-Cochrane, E.: 'Demonstrating Rigor Using Thematic Analysis: A Hybrid Approach of Inductive and Deductive Coding and Theme Development', *International Journal of Qualitative Methods*, 2006, 5, (1), pp. 80-92
- [9] Gómez, E., Grachten, M., Hanjalic, A., Janer, J., Jordà, S., Julià, C.F., Liem, C.C.S., Martorell, A., Schedl, M., and Widmer, G.: 'PHENICX: Performances as Highly Enriched and Interactive Concert Experiences', in: SMC Sound and Music Computing Conference. Stockholm, Sweden; 2013.
- [10] Inskip, C., MacFarlane, A., and Rafferty, P.: 'Upbeat and quirky, with a bit of a build: interpretive repertoires in creative music search', in: 11th International Society for Music Information Retrieval Conference (ISMIR 2010). Utrecht, The Netherlands; 2010, pp. 655-660.
- [11] Kolb, B.: 'You Call This Fun? Reactions of Young First-time Attendees to a Classical Concert', *Music & Entertainment Industry Educators Association (MEIEA) Journal*, 2000, 1, (1), pp. 13-28
- [12] Laplante, A.: 'Users' relevance criteria in music retrieval in everyday life: an exploratory study', in: 11th International Society for Music Information Retrieval Conference (ISMIR 2010). Utrecht, The Netherlands; 2010, pp. 601-606.
- [13] Lee, J.H.: 'Analysis of user needs and information features in natural language queries seeking music information', *Journal of the Association for Information Science and Technology*, 2010, 61, (5), pp. 1025-1045.
- [14] Lee, J.H., and Cunningham, S.J.: 'The impact (or non-impact) of user studies in music information retrieval', *Journal of Intelligent Information Systems*, 2013, 41, (3), pp. 499-521.
- [15] Liem, C.C.S., Mueller, M., Eck, D., Tanetakis, G. and Hanjalic, A.: 'The need for music information retrieval with user-centered and multimodal strategies', in: 1st International ACM Workshop on Music Information Retrieval with User-Centered Multimodal Strategies (MIRUM 2011). Scottsdale, AZ, USA; 2011, pp 1-6.
- [16] Liem, C.C.S., Van Der Sterren, R., Van Tilburg, M., Sarasúa, Á., Bosch, J.J., Melenhorst, M.S., Gómez, E., and Hanjalic, A.: 'Innovating the Classical Music Experience in the PHENICX Project: Use Cases and Initial User Feedback', in: 1st International Workshop on Interactive Content Consumption (WSICC) at EuroITV. Como, Italy; 2013.
- [17] Pitts, S.E., Dobson, M.C., Gee, K., and Spencer, C.P.: 'Views of an audience: Understanding the orchestral concert experience from player and listener perspectives', *Journal of Audience and Reception studies*, 2013, 10, (2), pp. 65-95
- [18] Radbourne, J., Johanson, K., Glow, H., and Tabitha, W.: 'The Audience Experience: Measuring Quality in the Performing Arts', *International Journal of Arts Management*, 2009, 11, (3), pp. 16-29
- [19] Roose, H.: 'Many-Voiced or Unisono?: An Inquiry into Motives for Attendance and Aesthetic Dispositions of the Audience Attending Classical Concerts', *Acta Sociologica*, 2008, 51, (3), pp. 237-253
- [20] Schedl, M., Flexer, A., and Urbano, J.: 'The neglected user in music information retrieval research', *Journal of Intelligent Information Systems*, 2013, 41, (3), pp. 523-539.
- [21] Venkatesh, V., Thong, J.Y.L., and Xin, X.: 'Consumer Acceptance and Use of Information Technology: Extending the Unified Theory of Acceptance and Use of Technology', *MIS Quarterly*, 2012, 36, (1), pp. 157-178
- [22] Wiering, F.: 'Meaningful music retrieval', in: 1st Workshop on the Future of MIR (f(MIR)). Kobe, Japan; 2009.

Oral Session 7

Performance

ANALYSIS OF EXPRESSIVE MUSICAL TERMS IN VIOLIN USING SCORE-INFORMED AND EXPRESSION-BASED AUDIO FEATURES

Pei-Ching Li¹ Li Su² Yi-Hsuan Yang² Alvin W. Y. Su¹

¹ SCREAM Lab., Department of CSIE, National Cheng-Kung University, Taiwan

² MAC Lab., CITI, Academia Sinica, Taiwan

p78021015@mail.ncku.edu.tw, lisu@citi.sinica.edu.tw

yang@citi.sinica.edu.tw, alvinsu@mail.ncku.edu.tw

ABSTRACT

The manipulation of different interpretational factors, including dynamics, duration, and vibrato, constitutes the realization of different expressions in music. Therefore, a deeper understanding of the workings of these factors is critical for advanced expressive synthesis and computer-aided music education. In this paper, we propose the novel task of automatic expressive musical term classification as a direct means to study the interpretational factors. Specifically, we consider up to 10 expressive musical terms, such as *Scherzando* and *Tranquillo*, and compile a new dataset of solo violin excerpts featuring the realization of different expressive terms by different musicians for the same set of classical music pieces. Under a score-informed scheme, we design and evaluate a number of note-level features characterizing the interpretational aspects of music for the classification task. Our evaluation shows that the proposed features lead to significantly higher classification accuracy than a baseline feature set commonly used in music information retrieval tasks. Moreover, taking the contrast of feature values between an expressive and its corresponding non-expressive version (if given) of a music piece greatly improves the accuracy in classifying the presented expressive one. We also draw insights from analyzing the feature relevance and the class-wise accuracy of the prediction.

1. INTRODUCTION

The expressive meaning of music is generally related to two inter-dependent factors: the *structure* established by the composer (e.g., mode, pitch, or dissonance) and the *interpretation* of the performer (e.g., expression) [21]. Glenn Gould could phrase the *trills* in a way different from other pianists. Mozart's *Grazioso* should be interpreted unlike to Brahms'. Although the interplay between the structural and interpretational factors makes it difficult to characterize musical expressiveness from audio signals, it has been

pointed out that such analysis is valuable in emerging applications such as automatic music transcription, computer-aided music education, or expressive music synthesis [2, 4, 7, 19]. Accordingly, computational analysis of the interpretational aspects in music expression has been studied for a while. For example, Bresin *et al.* analyzed the statistical behaviors of *legato* and *staccato* played with 9 expressive adjectives (not expressive musical terms) [3]. Grachten *et al.* made both predictive and explanatory modeling on the dynamic markings (e.g., *f*, *p*, *ff*, and *crescendo*) [10]. Ramirez *et al.* considered an approach of evolutionary computing for general timing and energy expressiveness [18]. Marchini *et al.* analyzed the performance of string quartets by the following three terms: *mechanical*, *normal* and *exaggerated* [14]. Recently, Rodà *et al.* further considered expressive constants as affective dimensions of music [20]. Related works also include the identification of performers, singers and instrument playing techniques in the context of musical expression [1, 6, 12, 15].

To model specific aspects of the complicated music expression quantitatively, a machine learning based approach is usually taken. Given an audio input, features are extracted to characterize the interpretational aspects of music, such as the dynamics, tempo and vibrato [3, 9, 12, 14].¹ If the symbolic or score data such as the MIDI or MusicXML are available, one can further introduce more structural aspects including tonality, pitch, note duration and measure, amongst others [10, 15, 16]. In [14], the synchronized audio, score and even motion data are utilized to generate 4 sets of features, including sound level, note lengthening, vibrato extent and bow velocity, in an attempt to reveal human behaviors while playing the instrument or indicate the structural information of music. This way, the features investigated have music meanings, and can be adopted for specific applications such as the prediction and the generation of expressive performances [10, 18].

Among all the objects of music expression, we notice that the *expressive musical terms* (EMT)² have garnered less attention in the literature, although they have been



© Pei-Ching Li, Li Su, Yi-Hsuan Yang, Alvin W. Y. Su. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Pei-Ching Li, Li Su, Yi-Hsuan Yang, Alvin W. Y. Su. "ANALYSIS OF EXPRESSIVE MUSICAL TERMS IN VIOLIN USING SCORE-INFORMED AND EXPRESSION-BASED AUDIO FEATURES", 16th International Society for Music Information Retrieval Conference, 2015.

¹ Here we assume that any real-world interpretation of an expressive musical term performed by a musician can be "atomized" into several (independent) factors such as dynamics, tempo, and vibrato.

² In this paper, the expressive musical term is defined as the Italian musical term which describes an emotion, feeling, image or metaphor, rather than merely an indication of tempo or dynamics. It includes, but not limited to the emotional terms (see Table 1).

Violin pieces	Measure	Expressions
W. A. Mozart - <i>Variationen</i>	1-24	<i>None, Scherzando, Tranquillo, Con Brio, Maestoso, Risoluto</i>
T. A. Vitali - <i>Chaconne</i>	1-9	<i>None, Scherzando, Affettuoso, Con Brio, Agitato, Cantabile</i>
G. Faure - <i>Elegie</i>	2-9	<i>None, Scherzando, Grazioso, Agitato, Espressivo, Cantabile</i>
P. I. Tchaikovsky - <i>String Quartet, No. 1, Mov. II</i>	1-16	<i>None, Affettuoso, Tranquillo, Con Brio, Cantabile, Risoluto</i>
M. Bruch - <i>Violin Concerto, No. 1, Mov. I</i>	6, 10 (solo, ad lib.)	<i>None, Affettuoso, Tranquillo, Agitato, Maestoso, Cantabile</i>
A. Vivaldi - <i>La primavera, Mov. I</i>	1-13	<i>None, Scherzando, Affettuoso, Grazioso, Con Brio, Risoluto</i>
A. Vivaldi - <i>La primavera, Mov. II</i>	2-11	<i>None, Grazioso, Agitato, Espressivo, Maestoso, Cantabile</i>
E. Elgar - <i>Salut d'Amour</i>	3-17	<i>None, Affettuoso, Grazioso, Agitato, Espressivo, Maestoso</i>
A. Vivaldi - <i>L'autunno, Mov. I</i>	1-13	<i>None, Tranquillo, Grazioso, Con Brio, Espressivo, Risoluto</i>
A. Vivaldi - <i>L'autunno, Mov. III</i>	1-29	<i>None, Scherzando, Tranquillo, Espressivo, Maestoso, Risoluto</i>

Table 1: The proposed dataset contains 10 different classical music pieces and each with 6 distinct expressions.

widely used in specifying expressions of classical music for hundreds of years. How the interpretational factors (dynamics, duration or vibrato) are taken for a musician to interpret the terms is still not well understood. This might be due to the lack of a dataset containing various interpretations for a fixed set of classical music pieces.

In this paper we address these issues, and particularly, focus on the classification of expressive musical terms in violin solo music. We compile a new dataset of solo violin excerpts featuring the realization of 10 expressive terms and 1 non-expressive term (e.g., no expression) by 11 different musicians for 10 classical music pieces (Section 2). After collecting the MIDI and MusicXML data for the music pieces, we design a number of dynamic-, duration- and vibrato-based features under a score-informed scheme (Section 3.2). Moreover, we also consider a baseline feature set comprising of standard audio features that can be computed without score information, such as the Mel-frequency cepstral coefficients (MFCCs), spectral flux, spectral centroid, and the zero-crossing rate (Section 3.1). As such features have been widely used in music information retrieval tasks like the classification of mood, genre or instruments [25], we want to know whether they are also useful for classifying the expressive musical terms. However, we should note that many of the baseline features do not bear clear music meanings as the proposed features do. In our experiments, we will evaluate the performance of these features for expressive musical term classification, and analyze the importance of such features (Section 4).

The dataset is referred to as the SCREAM-MAC-EMT dataset. For reproducibility and for calling more attention to this research problem, we have made the audio files of the recordings publicly available online.³

2. THE SCREAM-MAC-EMT DATASET

To find out how a violinist interprets the expressive musical terms, the scope of the music data, the difference in personal interpretation, and the suitability between the music piece and the musical term are all considered. We started by listing 20 typical violin pieces ranging across the Baroque, Classical, and Romantic eras, such as Vivaldi's *The Four Seasons*, Beethoven's *Spring*, and Schubert's *Ave Maria*, to name a few. Then, we consulted with 3 profes-

³ <https://sites.google.com/site/pclipatthy/scream-mac-emt-dataset>

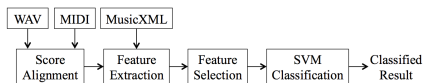


Figure 1: Flowchart of the proposed system.

sional violinists, who are active in classical music performance, to select 10 pieces from the list and assign 5 suitable expressive musical terms for each of them. The major criterion of selecting the music pieces, as it turns out, requires that an excerpt has a simple melody that can be effectively manipulated to exhibit different characteristics when being interpreted with different expressions.

The following 10 expressive terms are considered: *Tranquillo* (calm), *Grazioso* (graceful), *Scherzando* (playful), *Risoluto* (rigid), *Maestoso* (majestic), *Affettuoso* (affectionate), *Espressivo* (expressive), *Agitato* (agitated), *Con Brio* (bright), and *Cantabile* (like singing).⁴ In order to have a balanced dataset, we require that each expressive musical term is associated with 5 pieces. This is not easy, because not all of the 20 pieces can be interpreted with diverse expressions. Eventually, some compromises have to be made. For example, we chose *Maestoso* instead of *Cantabile* for Elgar's *Salut d'Amour*, although the former is somewhat awkward for this music piece. The resulting selection of the music pieces and the assigned expressions is shown in Table 1.

After selecting the music pieces, we recruited 11 professional violinists to perform them one by one in a real-world environment. In addition to the 5 assigned terms, every musician performed a non-expressive (denoted as *None*) version for each piece. Here, *None* means mechanical interpretation [14] by which the music is of constant dynamics, constant tempo and no vibrato. The dataset therefore contains 660 excerpts as there are 10 classical music pieces and each piece is interpreted by 6 different versions by all the 11 violinists. We have 110 excerpts of *None*, and 55 excerpts for each of the 10 expressions.

3. METHOD

Figure 1 shows the proposed system diagram. At the first stage of the system, the input audio signal is aligned with

⁴ For more information, see <http://www.musictheory.org.uk/res-musical-terms/italian-musical-terms.php>

Name	Abbreviation	Note-level aggregation	Song-level aggregation
Dynamics	D	M, Max, maxPos	M, S, C_M
Duration	ND, 1MD, 2MD, 4MD	—	M, S, C_M
	FPD	—	—, C_M
Vibrato rate	VR	M, S, $M\Delta$, $S\Delta$, Max, Min, Diff	M, S, C_M
Vibrato extent	VE	M, S, $M\Delta$, $S\Delta$, Max, Min, Diff	M, S, C_M
Global vibrato extent	GVE	—	M, S, C_M
Vibrato ratio	vibRatio	—	—

Table 2: Proposed features, the note-level and song-level aggregation methods.

its corresponding MIDI file in order to find the onset and offset positions and the pitch of each note in the audio signal. To do this, we adopt a chromagram-based audio-score alignment algorithm proposed in [23]. The positions of the bar lines are extracted from the MusicXML-formatted score sheets by using an XML parser.⁵ Then, to better characterize the attributes of the basic temporal elements (note or bar) of music, frame-level features are aggregated over time to generate note-level or bar-level features according to the desired segmentation. Furthermore, the note-level and bar-level features are aggregated again into a song-level representation, which allows us to map a variable-length sequence into a fixed-size feature vector that can be fed into a classifier. Finally, in the classification stage, we use radial-basis function (RBF) kernel Support Vector Machine (SVM) implemented by LIBSVM [5].

For the feature aggregation process from note-level (or bar-level) to song-level, we consider 3 different ways: (1) taking mean value over all notes in the excerpt (M), (2) taking standard deviation over all notes in the excerpt (S), and (3) taking the contrast of M between the expressive and its corresponding non-expressive version (C_M): $C_M = M_{\text{expressive}}/M_{\text{None}}$. C_M here is designed to “calibrate” the effect of *None*, which can be regarded as a baseline for the other 10 expressive musical terms. That is, C_M can somehow tell how different the expressive feature is from its non-expressive version. For the feature aggregation methods from frame-level to note-level, we will introduce them separately since they are different for each feature.

We introduce below the baseline feature set and the proposed feature set.

3.1 Baseline Features

The baseline features are a rich set of audio features covering dynamics, rhythm, tonal, and timbre. In particular, the baseline features are a rich set of temporal, spectral, cepstral and harmonic descriptors. It contains the mean and standard deviation of spectral centroid, brightness, spread, skewness, kurtosis, roll-off, entropy, irregularity, flatness, roughness, inharmonicity, flux, zero-crossing rate, low energy ratio, attack time, attack slope, dynamics and the mean and standard deviation of first-order temporal difference for all the above features, totaling $4 \times 17 = 68$ features. Besides, it involves the mean of fluctuation peak and centroid, tempo, pulse clarity and event density, generating 5 features; the mean and standard deviation of

mode and key clarity, resulting 4 features. Furthermore, it includes the mean and standard deviation of the 40-D MFCCs, Δ MFCCs (first-order temporal difference) and $\Delta\Delta$ MFCCs (second-order temporal difference), totaling $2 \times 120 = 240$ features. In sum, we have 317 features extracted by the MIRtoolbox (version 1.3.4) [13].

3.2 Proposed Features

3.2.1 Dynamic Features

The dynamics of each note is estimated from the short-time Fourier transform (STFT). Given a segmented note $x(n)$ and the Hanning window function $w(n)$, the STFT is represented as $X^w(n, k) = M^w(n, k) e^{j\Phi^w(n, k)}$, where $M^w(n, k)$ is the magnitude part, $\Phi^w(n, k)$ is the phase part, n is the time index, and k is the frequency index. The dynamic level function $D(n)$ is computed by the summation of the magnitude spectrogram over the frequency bins and is expressed in dB scale:

$$D(n) = 20 \log_{10} \left(\sum_k M(n, k) \right). \quad (1)$$

Three note-level dynamic features are computed from $D(n)$. Each of them are the mean value of $D(n)$ (D-M), the maximal value of $D(n)$ (D-Max) and the proportion of the maximum position to the note length (D-maxPos):

$$\text{maxPos} = \frac{\arg \max_n D(n)}{\text{length}(D(k))} \times 100\%. \quad (2)$$

D-maxPos therefore measures the time a note reaches its maximal energy from its beginning, normalized to the length of the note. All of these three note-level features are then aggregated to song-level by M, S, and C_M , totaling 9 features (see the second row of Table 2). For the $D(n)$ calculation, frames of 23ms (1014 samples) with an 82% overlap (832 samples), as used in [14], are adopted.

3.2.2 Duration Features

After score alignment and note segmentation, we take the following values as the features: the duration of every single note (ND), measure (1MD), two-measure segment (2MD), four-measure segment (4MD), and the full piece (FPD) (see the third row of Table 2). We expect that these features can capture the interpretation of local tempo variations measured by single notes, downbeats, and phrases. We take M and S on ND, 1MD, 2MD and 4MD to obtain song-level features. FPD itself is already a song-level feature so no aggregation is needed. Moreover, all of these

⁵ For more details about MusicXML, please refer to <http://www.musicxml.com/>

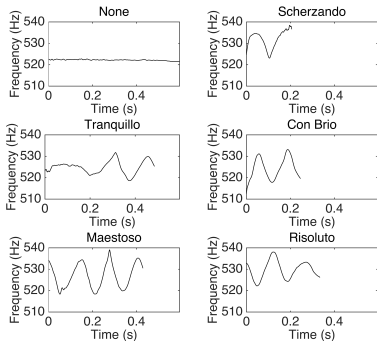


Figure 2: Pitch contours of the first crotchet (C5) of Mozart's *Variationen* with 6 expressions: *None*, *Scherzando*, *Tranquillo*, *Con Brio*, *Maestoso* and *Risoluto*.

five features are processed by C_M . Figure 2 shows examples where the same note (a crotchet C5) is interpreted in different ND for distinct expressions.

There are some more implementation details about the duration features. If a music piece has an incomplete measure in the beginning (e.g., Vivaldi's *La primavera Mov. I*) then the incomplete measure is merged into the next one and features are computed starting from the first complete measure. If the length of a phrase is not the multiple of the 2 or 4 measures then the remainders are combined as a group. Bruch's *Violin Concerto No. 1 Mov. I* (the 5th piece) is an unusual instance that has two *ad libitum* measures. In this case, 4MD is set at zero. In the parser process, a special part is to eliminate rests and ties because they do not have a unique sound. The former means an interval of silence and the latter has a curved line connecting to its previous note of the same pitch, indicating that they should be played as a single note.

3.2.3 Vibrato Features

Vibrato is an expressive manipulation of pitch corresponding to a frequency modulation of F0 (fundamental frequency) [17]. Because the vibrato is characterized by the rate and extent of the frequency modulation of F0, a precise estimation of the instantaneous pitch contour is needed. Since the frequency resolution in the STFT representation may not be high enough to represent the instantaneous frequency, we compute the instantaneous frequency deviation (IFD) [11] to estimate the instantaneous frequency:

$$\text{IFD}^w(n, k) = \frac{\partial \Phi^w}{\partial t} = \text{Im} \left(\frac{X^{D^w}(n, k)}{X^w(n, k)} \right), \quad (3)$$

where $D^w(n) = w'(n)$. Given the pitch of each note from the score, instantaneous frequency is computed by summing the IFD and the bin frequency of the bin which is nearest to the pitch frequency. Figure 2 also sketches examples of the vibrato contours. We can see large differences in both duration and vibrato among them. For the

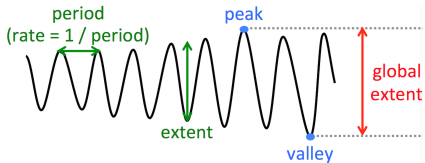


Figure 3: Illustration of vibrato rate, vibrato extent, and global vibrato extent in a single note.

IFD calculation, a window of 1025 samples at 44.1 kHz sampling rate and a hop size of 64 samples are applied.

After obtaining the vibrato contour of each note, we adopt a moving-average filter with length of one-hundredth of the note length to reduce the spurious variation of the pitch contour. The filter length is empirically set so as not to avoid much distortion and to remove high-frequent noise. Based on the smoothed pitch contour, we consider the *vibrato rate* (VR) and the *vibrato extent* (VE). The former means the reciprocal of the time duration of two consecutive peaks, while the latter means the frequency deviation between a peak and its nearby valley. Following [8], we require that a vibrato chain contains more than 3 points and VR is between 3 and 12 Hz; otherwise, the vibrato chain is excluded. For each note, we compute the mean, standard variation, mean of difference (M Δ), standard variation of difference (S Δ), maximum (Max), minimum (Min) and difference (Diff) between the maximal and minimal values of both VR and VE over all frames within a note [24]. These note-level features are also aggregated to song-level features by means of M, S, and C_M .

In addition, we consider a note-level feature called global vibrato extent (GVE), meaning the difference of the maximal peak value and the minimal valley value within a vibrato note as shown in Figure 3. GVE is also aggregated to song-level features through M, S, and C_M . Finally, we consider a song-level feature called vibrato ratio (vibRatio), defined as:

$$\text{vibRatio} = \frac{\# \text{ vibrato notes}}{\# \text{ notes in a violin piece}} \times 100\%. \quad (4)$$

When no vibrato note is detected or the ND is shorter than 125ms [14], the vibrato features are set at zero.

3.3 Feature Selection and Classification

To evaluate the importance of the adopted features in our task, we perform feature selection on both the baseline and the proposed feature sets. Here, the ReliefF routine of the MATLAB statistics toolbox⁶ is employed in the feature selection process [22]. In the training process, ReliefF sorts the features in descending order of relevance (importance). Then, the top- n' most relevant features are taken for SVM modeling. The optimal feature number n_{opt} which results in the best accuracy is obtained by brute-force searching.

⁶ <http://www.mathworks.com/products/>

Baseline	n	n_{opt}	c	γ	ACC
	317	107	1	2^{-8}	0.473
Proposed	without C_M				ACC
	n	n_{opt}	c	γ	ACC
Dynamics	6	6	$16 \cdot 2^{-6}$	0.318	0.398
Duration	9	8	$16 \cdot 2^{-4}$	0.331	0.380
Vibrato	31	6	$256 \cdot 2^{-6}$	0.264	0.258
All	46	22	$4 \cdot 2^{-6}$	0.425	0.531
Fusion	363	148	1	2^{-8}	0.589

Table 3: Performance of the baseline and the proposed feature sets. ‘All’ indicates the combination of dynamics, duration and vibrato; ‘fusion’ represents the combination of baseline and ‘all’. n and n_{opt} are the original and the optimized number of features respectively; c and γ are SVM parameters; ACC indicates the average accuracy.

The RBF-kernel SVM is adopted for classification. Since the dataset is recorded by 11 violinists, we simply take 11-fold cross validation, by using the data of 10 violinists as the training set and the other as the testing set. Then the feature selection is performed in each fold of the cross validation individually. After sending the top- n_{opt} most relevant features into classification, the resulting performance is obtained from optimizing the parameters c and γ of the SVM. In this work, the SVM parameters are set according to the highest average classification accuracy across the 11 folds. In the future, we will consider other data splitting settings, for example using an independent held-out set for parameter tuning.

In our classification experiment, we exclude the case of *None* and consider a 10-class multi-class classification problem, because the calculation of C_M aggregation method requires that the non-expressive version is known *a priori*. The classification accuracy of random guess is 0.152 on average.

As we want to find out the relevant interpretational factors, we only report below the results obtained by the top- n_{opt} relevant features selected by ReliefF.

4. EXPERIMENT RESULTS

4.1 Overview

Table 3 lists the original feature number n , the optimal feature number n_{opt} , the average accuracy (the ratio of true positives and the number of data) computed over the 11 folds, and the corresponding optimal c and γ for each experimental setting. The upper part of the table shows the result of the baseline feature set, where ReliefF selects $n_{opt} = 107$ out of 317 features and achieves an accuracy of 0.473. From the lower part of the table, when C_M aggregation method is considered, the proposed feature set achieves an accuracy of 0.531 when choosing $n_{opt} = 36$ out of 69 features, showing a significant improvement from the baseline feature set as validated by a one-tailed t-test ($p < 0.05$, d.f. = 20). Finally, after fusing the baseline features and all the proposed features, the average accuracy comes to 0.589, using $n_{opt} = 68$ out of 386 features.

#	Baseline	Proposed (‘all’)	Fusion
1	24th MFCC-M	4MD- C_M	vibRatio
2	18th MFCC-S	vibRatio	24th MFCC-M
3	26th MFCC-M	D-Max- C_M	ND-M
4	31st MFCC-M	D-M- C_M	18th MFCC-S
5	25th MFCC-S	FPD- C_M	VR-Min-M
6	15th MFCC-S	ND-M	31st MFCC-M
7	21st MFCC-S	D-maxPos-M	26th MFCC-M
8	31st MFCC-S	VR-Min-M	4MD- C_M
9	9th MFCC-S	1MD- C_M	25th MFCC-S
10	entropy-S Δ	2MD- C_M	9th MFCC-S
11	17th MFCC-S	FPD	FPD- C_M
12	24th MFCC-S	2MD-M	24th MFCC-S
13	16th MFCC-S	1MD-M	15th MFCC-S
14	23rd MFCC-M	ND- C_M	23rd MFCC-M
15	22nd MFCC-S	VR-M-M	31st MFCC-S
16	15th MFCC-M	4MD-S	D-maxPos-M
17	30th MFCC-M	4MD-M	16th MFCC-S
18	10th MFCC-S	D-maxPos-S	21st MFCC-S
19	16th MFCC-M	D-maxPos- C_M	10th MFCC-S
20	29th MFCC-S	D-M-M	entropy-S Δ

Table 4: The first 20 ranked features of the feature sets.

4.2 The contrast value C_M

Table 3 also shows how important using the contrast between the expressive and non-expressive version improves the performance. Comparing the left-hand side (without C_M) and the right-hand side (with C_M) of the table, using C_M constantly improves the average accuracy. Salient improvement can be observed for dynamic features ($p < 0.05$) and duration features ($p \geq 0.05$), implying that the change of dynamics, note duration, downbeat or phrase might be important interpretation factors when comparing the expressive and non-expressive performance. The improvement is not significant for vibrato ($p > 0.5$), possibly because the ratio of strong vibrato (expressive) and “almost no vibrato” (non-expressive) is not a stable feature. Table 3 also shows that using C_M on the proposed (‘all’) and the fusion feature sets leads to significant improvement for both cases ($p < 0.005$). Taking the contrast of feature values between expressive and non-expressive performance seems to be critical in modeling musical expression.

4.3 Feature importance analysis

Table 4 lists the top-20 relevant features for the baseline, proposed (‘all’) and fusion feature sets. The list is generated by summing the rank of each feature over the results of 11 folds, and by sorting the summarized rank again.

From the leftmost column, we can see that most of the relevant features in the baseline set are MFCCs. Despite its accuracy is inferior to the proposed features, this result shows the generality of MFCCs in audio classification.

From the middle column, we see that the top-20 proposed features include 11 duration features, 6 dynamic ones, and 3 vibrato ones. Over half of them are duration features. However, we note that the second feature is about vibrato (vibRatio) and the next two are both dynamic features (D-Max- C_M and D-M- C_M). It is not trivial to conclude that which factor is the most relevant. Dynamics, duration and vibrato all have contribution on music inter-

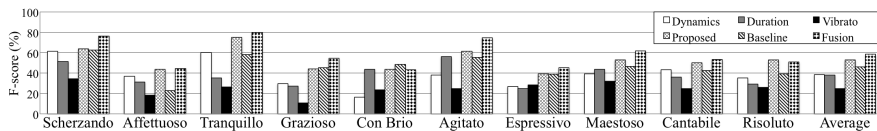


Figure 4: Average accuracy (in F-scores) of expression classification using individual feature sets.

	predicted class										F-score
	Sc	Af	Tr	Gr	Co	Ag	Es	Ma	Ca	Ri	
<i>Scherzando</i>	45	0	2	1	3	2	0	0	0	2	0.763
<i>Affettuoso</i>	1	22	3	4	3	3	4	4	8	3	0.444
<i>Tranquillo</i>	1	1	46	3	0	0	0	1	3	0	0.800
<i>Grazioso</i>	2	3	2	31	4	0	6	3	4	0	0.544
<i>Con Brio</i>	8	4	0	0	25	1	2	0	3	12	0.431
<i>Agitato</i>	0	1	0	1	2	43	4	4	0	0	0.748
<i>Espressivo</i>	1	0	1	9	2	5	23	6	5	3	0.451
<i>Maestoso</i>	0	2	3	4	1	5	4	34	1	1	0.618
<i>Cantabile</i>	1	9	3	5	2	1	4	1	29	0	0.532
<i>Risoluto</i>	4	2	0	1	19	0	0	2	1	26	0.510

Table 5: Confusion matrix of musical term classification using the fusion feature set.

pretation in various perspective. What this list provides is the signal-level details useful in synthesizing expressive music, or in education software for teaching expressive performance, something that cannot be achieved using the baseline features such as the MFCCs.

Finally, from the rightmost column we find that the top-20 fusion features contain a blending of the baseline and the proposed features. This shows that the two feature sets are indeed complementary, and it is advisable to exploit both of them if classification accuracy is of major concern.

4.4 Class-wise performance

Table 5 illustrates the confusion matrix of the fusion feature set summarized over the 11-fold outputs. In this confusion matrix, rows correspond to the actual class and columns correspond to the predicted class. The column on the right of the confusion matrix lists the average F-score, which is the harmonic mean of precision and recall.

We see that *Scherzando*, *Tranquillo* and *Agitato* attain relatively high F-scores because the first two have lighter dynamics than other expressions and the last one has shorter duration in most cases, all are fairly easy to be recognized. Interestingly, the low F-scores and the high confusion between the two pairs *Affettuoso*/*Grazioso* and *Cantabile*/*Espressivo* clearly reveal their semantic similarity. The most serious confusion occurs between *Risoluto* and *Con Brio*, perhaps due to their similar tempo and dynamics; the slightly difference of vibrato extent between them is not discriminated in our system, unfortunately.

Figure 4 shows the class-wise F-scores obtained by different feature sets. From the first three feature sets, we can find that using dynamic features outperforms other two for six expressions. Using duration features attains the best results for *Agitato*, *Con Brio* and *Maestoso*; the first two tend to use relatively fast tempo and the last one is prone to use a little slow and stable tempo. Lastly, we see that vi-

brato features perform slightly better than dynamic and duration features for *Espressivo*, possibly because *Espressivo* is similar to *Cantabile* in dynamic features and is similar to *Grazioso* in duration features.

Comparing the baseline to the proposed (‘all’) feature sets, the baseline feature set performs better only for *Grazioso* and *Con Brio*. The fusion set generally improves F-scores for all expressions except for *Con Brio*. For all settings, the four expressions, *Affettuoso*, *Grazioso*, *Espressivo* and *Cantabile*, are not easy to be distinguished from each other due to their similar meaning.

5. CONCLUSION AND FUTURE WORK

In this study, we have presented a method for analyzing the interpretational factors of expressive musical terms implemented on a new dataset comprising of rich expressive interpretations of violin solos. The proposed features, motivated from the basic understanding of dynamics, duration, vibrato, and the information of score, give better performance than the standard feature set in classifying expressive musical terms. Particularly, the contrast of feature values between expressive and non-expressive performance is found critical in modeling musical expression. The importance of the features is also reported. This provides insights into the design of new expression-based features, which may include features for the possible glissando between two adjacent notes, or the variation of the note/measure duration proportion with respect to its measure/excerpt. For future work, we will consider to expand the dataset, to experiment with other features and machine learning techniques, and to devise a mechanism that does not require a non-expressive reference to compute the contrast values.

6. ACKNOWLEDGMENTS

The authors would like to thank the following three professional violinists for consulting: Chia-Ling Lin (Doctor of Musical Arts, City University of New York); concertmaster of Counterpoint Ensemble, Liang-Chun Chou (Master of Music, Manhattan School of Music); concertmaster of Tainan Symphony Orchestra, and Hsin-Yi Su (Master of Music, New England Conservatory of Music); major violin performance of Tainan Symphony Orchestra). We are also grateful to the 11 professional violinists for their contribution to the development of the new dataset. The paper is partially funded by the Ministry of Science and Technology of Taiwan, under contracts MOST 103-2221-E-006-140-MY3 and 102-2221-E-001-004-MY3, and the Academia Sinica Career Development Award.

7. REFERENCES

- [1] J. Abeßer, H. Lukashevich, and G. Schuller. Feature-based extraction of plucking and expression styles of the electric bass guitar. In *ICASSP*, pages 2290–2293, 2010.
- [2] M. Barthet, P. Depalle, R. Kronland-Martinet, and S. Ystad. Analysis-by-synthesis of timbre, timing, and dynamics in expressive clarinet performance. *Music Perception*, 28(3):265–278, 2011.
- [3] R. Bresin and G. U. Battel. Articulation strategies in expressive piano performance analysis of legato, staccato, and repeated notes in performances of the andante movement of mozart’s sonata in g major (k 545). *Journal of New Music Research*, 29(3):211–224, 2000.
- [4] A. Camurri, G. Volpe, G. De Poli, and M. Leman. Communicating expressiveness and affect in multimodal interactive systems. *IEEE Multimedia*, 12(1):43–53, 2005.
- [5] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology*, 2(3):27:1–27:27, 2011.
- [6] J. Charles. *Playing Technique and Violin Timbre: Detecting Bad Playing*. PhD thesis, Dublin Institute of Technology, 2010.
- [7] R. L. De Mantaras. Playing with cases: Rendering expressive music with case-based reasoning. *AI Magazine*, 33(4):22, 2012.
- [8] A. Friberg, E. Schoonderwaldt, and P. N. Juslin. CUEX: An algorithm for automatic extraction of expressive tone parameters in music performance from acoustic signals. *Acta acustica united with acustica*, 93(3):411–420, 2007.
- [9] R. Gang, G. Bocko, J. Lundberg, S. Roessner, D. Headlam, and M. F. Bocko. A real-time signal processing framework of musical expressive feature extraction using MATLAB. In *ISMIR*, pages 115–120, 2011.
- [10] M. Grachten and G. Widmer. Linear basis models for prediction and analysis of musical expression. *Journal of New Music Research*, 41(4):311–322, 2012.
- [11] S. Hainsworth and M. Macleod. Time frequency re-assignment: A review and analysis. Technical report, Cambridge University Engineering Department, 2003.
- [12] N. Kroher and E. Gómez. Automatic singer identification for improvisational styles based on vibrato, timbre and statistical performance descriptors. In *ICMC-SMC*, 2014.
- [13] O. Lartillot and P. Toivainen. A matlab toolbox for musical feature extraction from audio. In *DAFx*, 2007.
- [14] M. Marchini, R. Ramirez, P. Papiotis, and E. Maestre. The sense of ensemble: a machine learning approach to expressive performance modelling in string quartets. *Journal of New Music Research*, 43(3):303–317, 2014.
- [15] M. Molina-Solana, J. L. Arcos, and E. Gómez. Using expressive trends for identifying violin performers. In *ISMIR*, pages 495–500, 2008.
- [16] K. Okumura, S. Sako, and T. Kitamura. Stochastic modeling of a musical performance with expressive representations from the musical score. In *ISMIR*, pages 531–536, 2011.
- [17] E. Prame. Vibrato extent and intonation in professional western lyric singing. *The Journal of the Acoustical Society of America*, (102):616–621, 1997.
- [18] R. Ramirez, E. Maestre, and X. Serra. A rule-based evolutionary approach to music performance modeling. *Evolutionary Computation, IEEE Transactions on*, 16(1):96–107, 2012.
- [19] C. Raphael. Representation and synthesis of melodic expression. In *IJCAI*, pages 1474–1480, 2009.
- [20] A. Rodà, S. Canazza, and G. De Poli. Clustering affective qualities of classical music: beyond the valence-arousal plane. *Affective Computing, IEEE Transactions on*, 5(4):364–376, 2014.
- [21] S. Vieillard, M. Roy, and I. Peretz. Expressiveness in musical emotions. *Psychological research*, 76(5):641–653, 2012.
- [22] M. R. Šikonja and I. Kononenko. Theoretical and empirical analysis of ReliefF and RReliefF. *Machine Learning*, 53:23–69, 2003.
- [23] T.-M. Wang, P.-Y. Tsai, and A. W. Y. Su. Note-based alignment using score-driven non-negative matrix factorisation for audio recordings. *IET Signal Processing*, 8:1–9, February 2014.
- [24] L. Yang, E. Chew, and K. Z. Rajab. Vibrato performance style: A case study comparing erhu and violin. In *ICMMR*, 2013.
- [25] Y.-H. Yang and H. H. Chen. Ranking-based emotion recognition for music organization and retrieval. *IEEE Trans. Audio, Speech & Lang. Processing*, 19(4):762–774, 2011.

SPECTRAL LEARNING FOR EXPRESSIVE INTERACTIVE ENSEMBLE MUSIC PERFORMANCE

Guangyu Xia Yun Wang Roger Dannenberg Geoffrey Gordon

School of Computer Science, Carnegie Mellon University, USA

{gxia, yunwang, rbd, ggordon}@cs.cmu.edu

ABSTRACT

We apply machine learning to a database of recorded ensemble performances to build an artificial performer that can perform music expressively in concert with human musicians. We consider the piano duet scenario and focus on the interaction of expressive timing and dynamics. We model different performers' musical expression as co-evolving time series and learn their interactive relationship from multiple rehearsals. In particular, we use a spectral method, which is able to learn the correspondence not only between different performers but also between the performance past and future by reduced-rank partial regressions. We describe our model that captures the intrinsic interactive relationship between different performers, present the spectral learning procedure, and show that the spectral learning algorithm is able to generate a more human-like interaction.

1. INTRODUCTION

Ensemble musicians achieve shared musical interpretations when performing together. Each musician performs expressively, deviating from a mechanical rendition of the music notation along the dimensions of pitch, duration, tempo, onset times, and others. While creating this musical interpretation, musicians in an ensemble must listen to other interpretations and work to achieve an organic, coordinated whole. For example, expressive timing deviations by each member of the ensemble are constrained by the overall necessity of ensemble synchronization. In practice, it is almost impossible to achieve satisfactory interpretations on the first performance. Therefore, musicians spend time in rehearsal to become familiar with the interpretation of each other while setting the "communication protocols" of musical expression. For example, when should each musician play rubato, and when should each keep a steady beat? What is the desired trend and balance of dynamics? It is important to notice that these protocols are usually complex and implicit in the sense that they are hard to express via explicit rules. (Musicians in a large ensemble even need a conductor to help set the protocols.) However, musicians are able to learn these protocols very effectively. After a few re-

hearsals, they are prepared to handle new situations that do not even occur in rehearsals, which indicates that the learning procedure goes beyond mere memorization.

Although many studies have been done on musical expression in solo pieces, the analysis of interactive ensemble music performance is relatively new and has mainly focused on mechanisms used for synchronization, including gesture. Ensemble human-computer interaction is still out of the scope of most *expressive performance* studies, and the interaction between synchronization and individual expressivity is poorly understood. From the synthesis perspective, though *score following and automatic accompaniment* have been practiced for decades, many researchers still refer to this as the "score following" problem, as if all timing and performance information derives from the (human) soloist and there is no performance problem. Even the term "automatic accompaniment" diminishes the complex collaborative role of performers playing together by suggesting that the (human) soloist is primary and the (computer) accompanist is secondary. In professional settings, even piano accompaniment is usually referred to as "collaborative piano" to highlight its importance. To successfully synthesize interactive music performance, all performers should be equal with respect to musical expression, including the artificial performers.

Thus, there is a large gap between music practice and computer music research on the topic of expressive interactive ensemble music performance. We aim to address this gap by mimicking human rehearsals, i.e., learn the communication protocols of musical expression from rehearsal data. For this paper, we consider the piano duet scenario and focus on the interaction of expressive timing and dynamics. In other words, our goal is to build an artificial pianist that can interact with a human pianist expressively, and is capable of responding to the musical nuance of the human pianist.

To build the artificial pianist, we first model different performers' musical expression as co-evolving time series and design a function approximation to reveal the interactive relationship between the two pianists. In particular, we assume musical expression is related to hidden mental states and characterize the piano duet performance as a *linear dynamic system* (LDS). Second, we learn the parameters of the LDS from multiple rehearsals using a spectral method. Third, given the learned parameters, the artificial pianist can generate an expressive performance by interacting with a human pianist. Finally, we conduct evaluation by comparing the computer-generated performances with human performances. At the same time, we



© Guangyu Xia, Yun Wang, Roger Dannenberg, Geoffrey Gordon. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Guangyu Xia, Yun Wang, Roger Dannenberg, Geoffrey Gordon. "Spectral Learning for Expressive Interactive Ensemble Performance", 16th International Society for Music Information Retrieval Conference, 2015.

inspect how training set size and the performer’s style affect the results.

The next section presents related work. Section 3 describes the model. Section 4 describes a spectral learning procedure. Section 5 shows the experimental results.

2. RELATED WORK

The related work comes from three different research fields: *Expressive Performance*, where we see the same focus of musical expression; *Automatic Accompaniment*, where we see the same application of human-computer interactive performance; and *Music Psychology*, where we see musicology insights and use them to help design better computational models. For detailed historical reviews of expressive performance and automatic accompaniment, we point the readers to [14] and [27], respectively. Here, we only review recent work that has strong connections to probabilistic modeling.

2.1 Expressive Performance

Expressive performance studies how to automatically render a musical performance based on a static score. To achieve this goal, probabilistic approaches learn the conditional distribution of the performance given the score, and then generate new performances by sampling from the learned models. Grindlay and Helmbold [9] use *hidden Markov models* (HMM) and learn the parameters by a modified version of the Expectation-Maximization algorithm. Kim et al. [13] use a *conditional random field* (CRF) and learn the parameters by stochastic gradient descent. Most recently, Flossmann et al. [7] use a very straightforward linear Gaussian model to generate the musical expression of every note independently, and then use a modification of the Viterbi algorithm to achieve a smoother global performance.

All these studies successfully incorporate musical expression with time-series models, which serve as good bases for our work. Notice that our work considers not only the relationship between score and performance but also the interaction between different performers. From an optimization point of view, these works aim to optimize a performance given a score, while our work aims to solve this optimization problem under the constraints created by the performance of other musicians. Also, we are dealing with a real-time scenario that does not allow any backward smoothing.

2.2 Automatic Accompaniment

Given a pre-defined score, automatic accompaniment systems follow human performance in real time and output the accompaniment by strictly following human’s tempo. Among them, Raphael’s Music Plus One [19] and IRCAM’s AnteScofo system [5] are very relevant to our work in the sense that they both use computational models to characterize the expressive timing of human musicians. However, the goal is still limited to temporal syn-

chronization; the computer’s musical expression in interactive performance is not yet considered.

2.3 Music Psychology

Most related work in Music Psychology, referred to as sensorimotor synchronization (SMS) and entrainment, studies adaptive timing behavior. Generally, these works try to discover common performance patterns and high-level descriptive models that could be connected with underlying brain mechanisms. (See Keller’s book chapter [11] for a comprehensive overview.) Though the discovered statistics and models are not “generative” and hence cannot be directly adopted to synthesize artificial performances, we can gain much musicology insight from their discoveries to design our computational models.

SMS studies how musicians tap or play the piano by following machine generated beats [15-18, 21, 25]. In most cases, the tempo curve of the machine is pre-defined and the focus is on how humans keep track of different tempo changes. Among them, Repp, Keller [21] and Mates [18] argue that adaptive timing requires error correction processes and use a “phase/period correction” model to fit the timing error. The experiments show that the error correction process can be decoupled into period correction (larger scale tempo change) and phase correction (local timing adjustment). This discovery suggests that it is possible to predict timing errors based on timing features on different scales.

Compared to SMS, entrainment studies consider more realistic and difficult two-way interactive rhythmic processes [1, 8, 10-11, 20, 22, 26]. Among them, Goebel [8] investigated the influences of audio feedback in a piano duet setting and claims that there exist bidirectional adjustments during full feedback despite the leader/follower instruction. Repp [20] does further analysis and discovers that the timing errors are auto-correlated and that how much musicians adapt to each other depends on the music context, such as melody and rhythm. Keller [11] claims that entrainment not only results in coordination of sounds and movements, but also of mental states. These arguments suggest that it is possible to predict the timing errors (and other musical expressions) by regressions based on different music contexts, and that hidden variables can be introduced to represent mental states.

3. MODEL SPECIFICATION

3.1 Linear Dynamic System (LDS)

We use a linear dynamic system (LDS), as shown in Figure 1, to characterize the interactive relationship between the two performers in the expressive piano duet. Here, $Y = [y_1, y_2, \dots, y_T]$ denotes the 2nd piano’s musical expression, $U = [u_1, u_2, \dots, u_T]$ denotes a combination of the 1st piano’s musical expression and score information, and $Z = [z_1, z_2, \dots, z_T]$ denotes the *hidden* mental states of the 2nd pianist that influence the performance. The key

idea is to reveal that the 2nd piano’s musical expression is not static. It is not only influenced by the 1st piano’s performance but also keeps its own character and continuity over time.

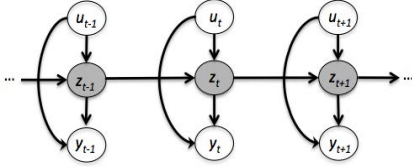


Figure 1. The graphical representation of the LDS, in which grey nodes represent hidden variables.

Formally, the evolution of the LDS is described by the following linear equations:

$$z_t = Az_{t-1} + Bu_t + w_t \quad w_t \sim \mathcal{N}(0, Q) \quad (1)$$

$$y_t = Cz_t + Du_t + v_t \quad v_t \sim \mathcal{N}(0, R) \quad (2)$$

Here, $y_t \in \mathbb{R}^2$ and its two dimensions correspond to expressive timing and dynamics, respectively, $u_t \in \mathbb{R}^l$, which is a much higher dimensional vector (we describe the design of u_t in detail in Section 3.3), and $z_t \in \mathbb{R}^n$, which is a relatively lower dimensional vector. A, B, C , and D are the main parameters of the LDS. Once they are learned, we can predict the performance of the 2nd piano based on the performance of the 1st piano.

3.2 Performance Sampling

Notice that the LDS is indexed by the discrete variable t . One question arises: should t represent note index or score time? Inspired by Todd’s work [23], we assume that musical expression evolves with score time rather than note indices, and therefore define t as score time. Since music notes have different durations, we “sample” the performed notes (of both the 1st piano and the 2nd piano) at the resolution of a half beat, as shown in Figure 2.

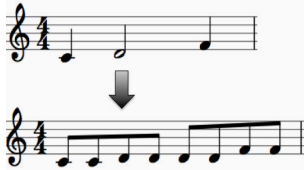


Figure 2. An illustration of performance sampling.

To be more specific, if a note’s starting time aligns with a half beat and its *inter-onset-interval* (IOI) is equal to or greater than one beat, we replace the note by a series of eighth notes, each having the same pitch, dynamic, and duration-to-IOI ratio as the original note. Note that we still play the notes as originally written; the sampled representation is only for learning and prediction.

3.3 Input Features Design

To show the design of u_t , we introduce an auxiliary notation $X = [x_1, x_2, \dots, x_T]$ to denote the raw score information and musical expression of the 1st piano and describe the mapping from X to each component of u_t in rest of this section. Note that u_t is based on sampled score and performance.

3.3.1 Score Features

High Pitch Contour: For the chords within a certain time window up to and including t , extract the highest-pitch notes and fit the pitches by a quadratic curve. Then, *high pitch contour* for t is defined as the coefficients of the curve. Formally:

$$\hat{\beta}_t^{high} \triangleq \underset{\beta}{\operatorname{argmin}} \sum_{i=0}^p (x_{t-p+i}^{highpitch} - \operatorname{quad}_{\beta}(t-p+i))^2$$

where p is a context length parameter and $\operatorname{quad}_{\beta}$ is the quadratic function parameterized by β .

Low Pitch Contour: Similar to *high pitch contour*, we compute $\hat{\beta}_t^{low}$ for *low pitch contour*.

Beat Phase: The relative location of t within a measure. Formally:

$$\operatorname{BeatPhase}_t \triangleq (t \bmod \operatorname{MeasureLen}) / \operatorname{MeasureLen}$$

3.3.2 The 1st Piano Performance Features

Tempo Context: Tempi of the p closest notes directly before t . This is a timing feature on a relatively large time scale. Formally:

$$\operatorname{TempoContext}_t \triangleq [x_{t-p}^{\operatorname{Tempo}}, x_{t-p+1}^{\operatorname{Tempo}}, \dots, x_{t-1}^{\operatorname{Tempo}}]^T$$

Here, the tempo of a note is defined as the slope of the least-squares linear regression between the performance onsets and the score onsets of q preceding notes.

Onsets Deviation Context: A description of how much the p closest notes’ onsets deviate from their tempo curves. Compared to the tempo context, this is a timing feature on a relatively small scale. Formally:

$$\operatorname{OnsetsDeviationContext}_t \triangleq [x_{t-p}^{\operatorname{OnsetsDeviation}}, x_{t-p+1}^{\operatorname{OnsetsDeviation}}, \dots, x_{t-1}^{\operatorname{OnsetsDeviation}}]^T$$

Duration Context: Durations of the p closest notes directly before t . Formally:

$$\operatorname{DurationContext}_t \triangleq [x_{t-p}^{\operatorname{Dur}}, x_{t-p+1}^{\operatorname{Dur}}, \dots, x_{t-1}^{\operatorname{Dur}}]^T$$

Dynamic Context: MIDI velocities of the p closest notes directly before t . Formally:

$$\operatorname{DynamicContext}_t \triangleq [x_{t-p}^{\operatorname{Vel}}, x_{t-p+1}^{\operatorname{Vel}}, \dots, x_{t-1}^{\operatorname{Vel}}]^T$$

The input feature, u_t , is a concatenation of the above features. We have also tried other features and mappings (e.g., rhythm context, phrase location, and down beat),

and finally picked the ones above through experimentation.

4. SPECTRAL LEARNING PROCEDURE

To learn the model, we use a spectral method, which is rooted in control theory [24] and then further developed in the machine learning field [2]. Spectral methods have proved to be both fast and effective in many applications [3][4]. Generally speaking, a spectral method learns hidden states by predicting the performance future from features of the past, but forcing this prediction to go through a low-rank bottleneck. In this section, we present the main learning procedure with some underlying intuitions, using the notation of Section 3.1.

Step 0: Construction of Hankel matrices

We learn the model in parallel for fast computation. In order to describe the learning procedure more concisely, we need some auxiliary notations. For any time series $S = [s_1, s_2, \dots, s_T]$, the “history” and “future” Hankel matrices are defined as follows:

$$S_H \stackrel{\text{def}}{=} \begin{pmatrix} s_1 & \dots & s_{T-d} \\ \vdots & \ddots & \vdots \\ s_{\frac{d}{2}} & \dots & s_{T-\frac{d}{2}-1} \end{pmatrix}, S_F \stackrel{\text{def}}{=} \begin{pmatrix} s_{\frac{d}{2}+1} & \dots & s_{T-\frac{d}{2}} \\ \vdots & \ddots & \vdots \\ s_d & \dots & s_{T-1} \end{pmatrix}$$

Also, the “one-step-extended future” and “one-step-shifted future” Hankel matrices are defined as follows:

$$S_F^+ \stackrel{\text{def}}{=} \begin{pmatrix} s_{\frac{d}{2}+1} & \dots & s_{T-\frac{d}{2}} \\ \vdots & \ddots & \vdots \\ s_{d+1} & \dots & s_T \end{pmatrix}, S_F^- \stackrel{\text{def}}{=} \begin{pmatrix} s_{\frac{d}{2}+2} & \dots & s_{T-\frac{d}{2}+1} \\ \vdots & \ddots & \vdots \\ s_{d+1} & \dots & s_T \end{pmatrix}$$

Here, d is an even integer indicating the size of a sliding window. Note that corresponding columns of S_H and S_F are “history-future” pairs within sliding windows of size d ; compared with S_F^+ , S_F^- is just missing the first row. We will use the Hankel matrices of both U and Y in the following steps.

Step 1: Oblique projections

If the true model is LDS, i.e., everything is linear Gaussian, the expected future observations can be expressed linearly by history observations, history inputs, and future inputs. Formally:

$$\mathbb{E}(Y_F | Y_H, U_H, U_F) = [\beta_{Y_H} \beta_{U_H} \beta_{U_F}] \begin{bmatrix} Y_H \\ U_H \\ U_F \end{bmatrix} \quad (3)$$

Here, $\beta = [\beta_{Y_H} \beta_{U_H} \beta_{U_F}]$ is the linear coefficient that could be solved by:

$$\hat{\beta} = [\hat{\beta}_{Y_H} \hat{\beta}_{U_H} \hat{\beta}_{U_F}] = Y_F \begin{bmatrix} Y_H \\ U_H \\ U_F \end{bmatrix}^\dagger \quad (4)$$

where \dagger denotes the Moore-Penrose pseudo-inverse. However, since in a real-time scenario the future input, U_F , is unknown, we can only partially explain future observations based on the history. In other words, we care

about the best estimation of future observations but just based on the history observations and inputs. Formally:

$$\hat{O}_F \stackrel{\text{def}}{=} \hat{\beta}_H \begin{bmatrix} Y_H \\ U_H \\ 0 \end{bmatrix} = [\hat{\beta}_{Y_H} \hat{\beta}_{U_H} 0] \begin{bmatrix} Y_H \\ U_H \\ 0 \end{bmatrix} \quad (5)$$

where \hat{O}_F is referred to as the oblique projection of Y_F “along” U_F and “onto” $\begin{bmatrix} Y_H \\ U_H \end{bmatrix}$. In this step, we also use the same technique to compute \hat{O}_F^+ and just throw out its first row to obtain \hat{O}_F^- .

Step 2: State estimation by singular value decomposition (SVD)

If we knew the true parameters of the LDS, the oblique projections and the hidden states would have the following relationship:

$$\hat{O}_F = \Gamma_f Z_f \stackrel{\text{def}}{=} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{\frac{d}{2}-1} \end{bmatrix} \begin{bmatrix} z_{\frac{d}{2}+1}^d, z_{\frac{d}{2}+2}^d, \dots, z_{T-\frac{d}{2}}^d \end{bmatrix} \quad (6)$$

$$\hat{O}_F^\pm = \Gamma_f^\pm Z_f^\pm \stackrel{\text{def}}{=} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{\frac{d}{2}-1} \end{bmatrix} \begin{bmatrix} z_{\frac{d}{2}+2}^d, z_{\frac{d}{2}+3}^d, \dots, z_{T-\frac{d}{2}+1}^d \end{bmatrix} \quad (7)$$

Intuitively, the information from the history observations and inputs “concentrate” on the nearest future hidden state and then spread out onto future observations. Therefore, if we perform SVD on the oblique projections and throw out small singular values, we essentially enforce a bottleneck on the graphical model representation, learning compact, low-dimensional states. Formally, let

$$\hat{O}_F = \mathcal{U} \Lambda \mathcal{V}^T \quad (8)$$

and delete small numbers in Λ and corresponding columns in \mathcal{U} and \mathcal{V} . Since LDS is defined up to a linear transformation, we could estimate the hidden states by:

$$\hat{\Gamma}_f = \mathcal{U} \Lambda^{\frac{1}{2}} \quad (9)$$

$$\hat{Z}_f = \hat{\Gamma}_f^\dagger \hat{O}_F \quad (10)$$

$$\hat{Z}_f^\pm = \hat{\Gamma}_f^\pm \hat{O}_F^\pm \quad (11)$$

Step 3: Parameter estimation

Once we have estimated the hidden states, the parameters can be estimated from the following two equations:

$$\hat{Z}_f^S = A \hat{Z}_f + B U_f^S + e_w \quad (12)$$

$$Y_f = C \hat{Z}_f + D U_f + e_v \quad (13)$$

Here, Y_f and U_f are the 1st rows of Y_F and U_F , i.e., $Y_f = [y_{\frac{d}{2}+1}^d, y_{\frac{d}{2}+2}^d, \dots, y_{T-\frac{d}{2}}^d]$, $U_f = [u_{\frac{d}{2}+1}^d, u_{\frac{d}{2}+2}^d, \dots, u_{T-\frac{d}{2}}^d]$. Similarly, U_f^S is the 1st row of U_F^S , i.e., $U_f^S = [u_{\frac{d}{2}+2}^d, u_{\frac{d}{2}+3}^d, \dots, u_{T-\frac{d}{2}+1}^d]$.

In summary, the spectral method does three regressions. The first two estimate the hidden states by oblique projections and SVD. The third one estimates the param-

ters. The oblique projections can be seen as de-noising the latent states by using past observations, while the SVD adds low-rank constraints. As opposed to maximum likelihood estimation (MLE), the spectral method is a method-of-moments estimator that does not need any random initialization or iterations. Also note that we are making a number of arbitrary choices here (e.g., using equal window sizes for history and future), not attempting to give a full description of how to use spectral methods. (See Van Overschee & De Moor’s book [24] for the details and variations of the learning methods.)

5. EXPERIMENTS

5.1 Dataset

We created a dataset [27] that contains three piano duets: *Danny Boy*, *Serenade* (by Schubert), and *Ashokan Farewell*. All pieces are in MIDI format and contain two parts: a monophonic 1st piano part and a polyphonic 2nd piano part. Each piece is performed 35 to 42 times in different musical interpretations by 5 to 6 pairs of musicians. (Each pair performs each piece of music 7 times.)

5.2 Methods for Comparison

We use three methods for comparison: linear regression, neural network, and the timing estimation often used in automatic accompaniment systems [6]. The first two methods use the same set of features as in the spectral methods, while the 3rd method does not contain any learning procedure and is considered as the baseline.

Linear regression: Referring to the notation in Section 3, the linear regression method simply solves the following equation:

$$Y = \beta U \quad (14)$$

Like the LDS, this method uses the performance of 1st piano to estimate that of the 2nd piano, but it does not use any hidden states or attempt to enforce self-consistency in the musical expression of the 2nd pianist’s performance.

Neural network: We use a simple neural network with a single hidden layer. The hidden layer consists of 10 neurons and uses rectified linear units (ReLU) to produce non-linearity; the single output neuron is linear. Denoting the activation of the hidden units by Z , the neural network represents the following relationship between U and Y :

$$Z = f(W_1 U + b_1) \quad (15)$$

$$Y = W_2 Z + b_2 \quad (16)$$

where

$$f(x) = \begin{cases} 0, & x < 0 \\ x, & x \geq 0 \end{cases} \quad (17)$$

The neural network is trained by the minibatch stochastic gradient descent (SGD) algorithm, using the mean absolute error as the cost function. The parameters of the neural network (W_1 , b_1 , W_2 , b_2) are initialized randomly, after

which they are tuned with 30 epochs of SGD. Each minibatch consists of one rehearsal. The learning rate decays from 0.1 to 0.05 in an exponential fashion during the training. We report the average absolute and relative errors across five runs with different random initializations on the test set.

This method can be seen as an attempt to improve the linear regression method using non-linear function approximation, but it also doesn’t consider the self-consistency in the musical expression of the 2nd pianist’s performance.

Baseline: The baseline method assumes that local tempo and dynamics are stable. For timing, it estimates a linear mapping between real time and score time by fitting a straight line to 4 recently performed note onsets of the 1st piano. This mapping is then used to estimate the timing of the next note of the 2nd piano. For dynamics, it uses the dynamics of the last performed note of the 1st piano as the estimator.

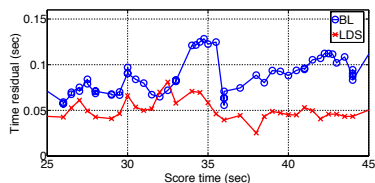


Figure 3. A local view of the absolute timing residuals of the LDS approach.

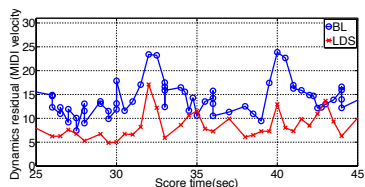


Figure 4. A local view of the absolute dynamics residuals of the LDS approach.

5.3 A Local View of the LDS Method

Figure 3 and Figure 4 show a local view of the expressive timing and dynamics cross-validation result, respectively, for *Danny Boy*. (To have a clear view, we just compare LDS with the baseline here. We show the results of all the methods on all the pieces later.) For both figures, the x-axis represents score time and the y-axis represents absolute residual between the prediction and human performance. Therefore, small numbers mean better results. The curve with circle markers represents the baseline approach, while the curve with “x” markers represents the LDS approach trained with only 4 randomly selected rehearsals of the *same* piece performed by *other* performers. We can see that the LDS approach performs much

better than the baseline approach with only 4 training rehearsals, which indicates that the algorithm is both accurate and robust.

5.4 A Global View of All Methods

The curves in the previous two figures are a measurement over different performances. If we average the absolute residual across an entire piece of music, we get a single number that describes a method’s performance for that piece. I.e., how much on average is the prediction of a method different from the human performance for each note? Figure 5 and Figure 6 show this average absolute residual for timing and dynamics, respectively, for all the methods and pieces combinations with different training set sizes.

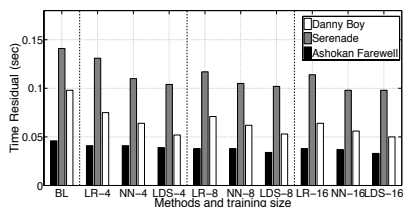


Figure 5. A global view of absolute timing residuals for all pieces and methods. (Smaller is better.)

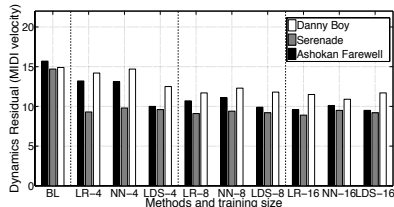


Figure 6. A global view of absolute dynamics residuals for all pieces and methods. (Smaller is better.)

In both figures, the x-axis represents different methods with different training set sizes, the y-axis represents the average absolute residual, and different colors represent different pieces. For example, the grey bar above the label “NN-4” in Figure 5 is the average absolute timing residual for *Serenade* by using the neural network approach with 4 training rehearsals.

We see that for expressive timing, both neural network and LDS outperform simple linear regression, and the LDS performs the best regardless of the music piece or training set size. This indicates that the constraint of preceding notes (self-consistency) captured by LDS is playing an important role in timing prediction. For expressive dynamics, the difference between different methods is less significant. We see no benefit by using a neural network. But when the training set size is small, LDS still

outperforms linear regression. (Which is quite interesting because LDS learns more parameters than linear regression.)

5.5 Performer’s Effect

Finally, we inspect whether there is any gain by training a performer-specific model. In other words, we only learn from the rehearsals performed by the *same* pair of musicians. Since each pair of musicians only performs 7 times for each piece, we randomly choose 4 from the 7 performances to make a fair comparison against the results in Figure 5 and Figure 6.

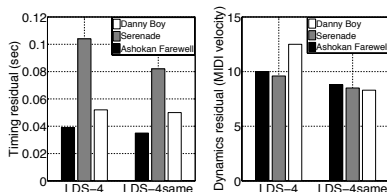


Figure 7. A global view of the performer-specific model.

Figure 7 shows a comparison between performer-specific model and different-performer model. In both sub-graphs, the bars above “LDS-4same” are the results for performer-specific model, while the bars above “LDS-4” are the same as in Figure 5 and Figure 6. Note that they are both cross-validation results and the only difference is the training set. We see that the performer-specific model achieves better results, especially when the different-performer model is not doing a good job.

6. CONCLUSIONS AND FUTURE WORK

In conclusion, we have applied a spectral method to learn the interactive relationship in expressive piano duet performances from multiple rehearsals. Compared to other methods, we have made better predictions based on only 4 rehearsals, and we have been able to further improve the results using a performer-specific model. Our best model is able to shrink the timing residual by nearly 60 milliseconds and shrink the dynamic residual by about 8 MIDI velocity units compared to the baseline algorithm, especially when the baseline algorithm behaves poorly.

In the future, we would like to incorporate some non-linear function approximations with the current graphical representation of the model. An ideal case would be to combine the dynamical system with a neural network, which calls for new spectral learning algorithms. Also, we would like to be more thorough in the evaluations. Rather than just inspecting the absolute difference between computer-generated performance and human performances, we plan to also compare computed-generated results with typical variation in human performances and use subjective evaluation.

7. REFERENCES

- [1] C. Bartlette, D. Headlam, M. Bocko, and G. Velickic, "Effect of Network Latency on Interactive Musical Performance," *Music Perception*, pp. 49–62, 2006.
- [2] B. Boots, *Spectral Approaches to Learning Predictive Representations* (No. CMU-ML-12-108). Carnegie Mellon Univ., School of Computer Science, 2012.
- [3] B. Boots and G. Gordon, "An Online Spectral Learning Algorithm for Partially Observable Nonlinear Dynamical Systems," *Proceedings of the National Conference on Artificial Intelligence*, 2011.
- [4] B. Boots, S. Siddiqi, and G. Gordon, "Closing the Learning-planning Loop with Predictive State Representations," *The International Journal of Robotics Research*, pp. 954-966, 2011.
- [5] A. Cont, "ANTESCOFO: Anticipatory Synchronization and Control of Interactive Parameters In Computer Music," *Proceedings of International Computer Music Conference*, pp. 33-40, 2011.
- [6] R. Dannenberg, "An Online Algorithm for Real-Time Accompaniment," *Proceedings of the International Computer Music Conference*, pp. 193-198, 1984.
- [7] S. Flossmann, M. Grachten, and G. Widmer, "Expressive Performance Rendering with Probabilistic Models," *Guide to Computing for Expressive Music Performance*, Springer, pp. 75–98, 2013.
- [8] W. Goebel and C. Palmer, "Synchronization of Timing and Motion Among Performing Musicians," *Music Perception*, pp. 427–438, 2009.
- [9] G. Grindlay and D. Helmbold, "Modeling, Analyzing, and Synthesizing Expressive Piano Performance with Graphical Models," *Machine Learning*, pp. 361-387, 2006.
- [10] M. Hove, M. Spivey, and L. Krumhansl, "Compatibility of Motion Facilitates Visuomotor Synchronization," *Journal of Experimental Psychology: Human Perception and Performance*, pp. 1525-1534, 2010.
- [11] P. Keller, "Joint Action in Music Performances," *Enacting Intersubjectivity: A Cognitive and Social Perspective to the Study of Interactions* Amsterdam, The Netherlands: IOS Press, pp. 205-221, 2008.
- [12] P. Keller, G. Knoblich, and B. Repp, "Pianists Duet Better When They Play with Themselves: On the Possible Role of Action Simulation in Synchronization," *Consciousness and Cognition*, pp. 102–111, 2007.
- [13] T. Kim, F. Satoru, N. Takuya, and S. Shigeki, "Polyhymnia: An Automatic Piano Performance System with Statistical Modeling of Polyphonic Expression and Musical Symbol Interpretation," *Proceedings of the International Conference on New Interfaces for Musical Expression*, pp. 96-99, 2011.
- [14] A. Kirke and E. R. Miranda, "A Survey of Computer Systems for Expressive Music Performance," *ACM Surveys* 42(1): Article 3, 2009.
- [15] E. Large and J. Kolen, "Resonance and the Perception of Musical Meter". *Connection Science*, pp. 177–208, 1994.
- [16] E. Large and C. Palmer, "Perceiving Temporal Regularity in Music," *Cognitive Science*, pp. 1–37, 2002.
- [17] E. Large and C. Palmer, "Temporal Coordination and Adaptation to Rate Change in Music Performance," *Journal of Experimental Psychology: Human Perception and Performance*, pp. 1292-1309, 2011.
- [18] J. Mates, "A Model of Synchronization of Motor Acts to a Stimulus Sequence: Timing and Error Correction," *Biological Cybernetics*, pp. 463–473, 1994.
- [19] C. Raphael, "Music Plus One and Machine Learning," *Proceedings of International Conference on Machine Learning*, pp. 21-28, 2010.
- [20] B. Repp and P. Keller, "Sensorimotor Synchronization with Adaptively Timed Sequences," *Human Movement Science*, pp. 423-456, 2008.
- [21] B. Repp and P. Keller, "Adaptation to Tempo Changes in Sensorimotor Synchronization: Effects of Intention, Attention, and Awareness," *Quarterly Journal of Experimental Psychology*, pp. 499-521, 2004.
- [22] G. Schöner, "Timing, Clocks, and Dynamical Systems," *Brain and Cognition*, pp. 31-51, 2002.
- [23] P. Todd, "A Connectionist Approach to Algorithmic Composition," *Computer Music Journal*, pp. 27-43, 1989.
- [24] P. Van Overschee and B. De Moor, *Subspace Identification for Linear Systems: Theory, Implementation, applications*. Kluwer Academic Publishers, 1996.
- [25] D. Vorberg and H. Schulze, "A Two-level Timing Model for Synchronization," *Journal of Mathematical Psychology*, pp. 56–87, 2002.
- [26] A. Wing, "Voluntary Timing and Brain Function: an Information Processing Approach," *Brain and Cognition*, pp. 7-30, 2002.
- [27] G. Xia and R. Dannenberg, "Duet Interaction: Learning Musicianship for Automatic Accompaniment," *Proceedings of the International Conference on New Interfaces for Musical Expression*, 2015.

SCORE-INFORMED ANALYSIS OF INTONATION AND PITCH MODULATION IN JAZZ SOLOS

Jakob Abeßer^{1,2} Estefanía Cano² Klaus Frieler¹
Martin Pfeleiderer¹ Wolf-Georg Zaddach¹

¹ Jazzomat Research Project, University of Music Franz Liszt, Weimar, Germany

² Semantic Music Technologies Group, Fraunhofer IDMT, Ilmenau, Germany

`jakob.abesser@idmt.fraunhofer.de`

ABSTRACT

The paper presents new approaches for analyzing the characteristics of intonation and pitch modulation of woodwind and brass solos in jazz recordings. To this end, we use score-informed analysis techniques for source separation and fundamental frequency tracking. After splitting the audio into a solo and a backing track, a reference tuning frequency is estimated from the backing track. Next, we compute the fundamental frequency contour for each tone in the solo and a set of features describing its temporal shape. Based on this data, we first investigate, whether the tuning frequencies of jazz recordings changed over the decades of the last century. Second, we analyze whether the intonation is artist-specific. Finally, we examine how the modulation frequency of vibrato tones depends on contextual parameters such as pitch, duration, and tempo as well as the performing artist.

1. INTRODUCTION

The personal styles of improvising jazz musicians can be described from various musical perspectives. There are several structural or syntactical aspects of the improvised melodic lines, which could be idiosyncratic for a certain musician, e.g., preferred pitches, intervals, scales, melodic contours, rhythms or typical patterns, licks, and formulas. These dimensions can be best explored using a symbolic representation, e.g., Western staff notation or MIDI. However, there are other important aspects, which define personal style and make it recognizable: *timbre* (sound characteristics such as roughness or breathiness), *micro-timing* (systematic deviations from the underlying metric structure), *dynamics* (the changes in intensity of tones or phrases), *intonation* (the pitch accuracy with respect to a given tone system), *articulation* (e.g., legato or staccato playing) and *pitch modulation* (the variation of the fundamental frequency within the duration of a tone). Symbolic

representation does not reveal information about timbre, intonation, and pitch modulation. Therefore, audio-level analysis of recorded improvisations is necessary to characterize those non-syntactical, expressive dimensions in order to get a comprehensive and exhaustive description of a personal style.

2. GOALS

Polyphonic music recordings exhibit strong spectral and temporal overlaps between harmonic components of different instrument sources. Hence, the transcription and analysis of the individual sound sources remain one of the most challenging tasks in Music Information Retrieval (MIR). We approach this task by using high-quality melody transcriptions provided by music experts as foundation for a score-informed audio analysis. In particular, we use score information for the source separation of the solo instrument from the audio mixture and for the frame-wise tracking of the fundamental frequency of each tone. Our main goal is to investigate, which intonation and modulation strategies are applied by woodwind and brass instrument players in jazz solos.

3. RELATED WORK

Various MIR publications investigate the *intonation* and *tuning* of music recordings, ranging from historic solo harpsichord recordings [5], over classical music recordings [11], to Non-Western music styles such as Carnatic and Hindustani music [17]. The tuning frequency of audio recordings is commonly estimated based on pitch frequencies [6], high-resolution interval histograms [17], or adjustable filterbanks [11]. Just intonation and equal temperament are generally used as reference tunings for the analysis of music performances. Lerch [11] points out that observed tuning deviations can have different reasons ranging from deviation of harmonic frequencies from the equal tempered scale to deviations due to non-equal temperament.

Most automatic music transcription algorithm aim at a symbolic representation of tone events, which are described by distinct onset times, durations, and constant pitches [15]. Some automatic melody extraction algorithms such as proposed in [16] and [6] include an estimation of the tone-wise contours of the fundamental frequency (f_0)



© Jakob Abeßer, Estefanía Cano, Klaus Frieler, Martin Pfeleiderer, Wolf-Georg Zaddach. Licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). **Attribution:** Jakob Abeßer, Estefanía Cano, Klaus Frieler, Martin Pfeleiderer, Wolf-Georg Zaddach. "Score-informed Analysis of Intonation and Pitch Modulation in Jazz Solos", 16th International Society for Music Information Retrieval Conference, 2015.

as well, which is an essential pre-processing step for analyzing the applied *frequency modulation techniques*. There are many studies on *vibrato* detection in audio recordings [14], particularly for singing voice [8,9,12]. Other publications deal with analyzing the deviation of f_0 contours from the target pitch [8] as well as with segmenting f_0 contours based on modulations such as vibrato and *pitch glides* [12] or *bendings* [10]. To the best knowledge of the authors, no publication so far analyzes intonation and modulation techniques in recorded jazz solos.

4. METHOD

Figure 1 gives an overview over our analysis approach, all processing steps are detailed in the following sections. Section 4.1 describes the dataset of jazz solo audio excerpts and transcriptions. Two separate score-informed analysis techniques are involved. At first, a *source separation* algorithm is performed (see Section 4.2), which separates the original audio recording into a solo track containing the improvising solo instrument and a backing track containing the accompanying band, i.e., the rhythm section (most often piano, double bass, and drums). The backing track is used to estimate the *reference tuning frequency* (see Section 4.4). The second step is the *tracking of frame-wise f_0 contours* for each note played by the solo instrument (see Section 4.3). Based on the extracted f_0 contours, we compute several *contour features* (see Section 4.5) to describe their temporal shape. In the experiments reported in Section 5, we analyze how these features depend on contextual parameters such as tone duration and pitch and whether these might be specific for the personal style.

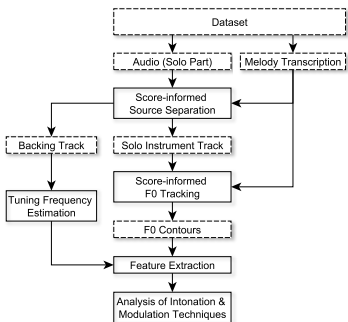


Figure 1: Proposed algorithm for score-informed analysis of tuning and modulation in improvised jazz solos.

4.1 Dataset & Melody Annotations

The dataset used in this publication is a subset of 207 jazz solos taken from the *Weimar Jazz Database*¹. Table 1 lists

¹ <http://jazzomat.hfm-weimar.de> (last accessed Juli 10, 2015)

all musicians in the dataset with their instrument, the number of solos N_S , and the total number of tones and f_0 contours N_N , respectively. The solos were manually annotated by musicology and jazz students based on excerpts from commercial audio recordings. The annotations include score-level melody transcription (MIDI pitch, tone onset, and duration) as well as additional annotation layers with respect to melody phrases, metric structure, chords, and modulation techniques. So far, the tone-wise annotations of modulation techniques are incomplete and only represent the most clear examples within the solos. Figure 2 gives an overview over the number of annotated tones per artist. In total, 87643 tones and f_0 contours are included in the dataset.

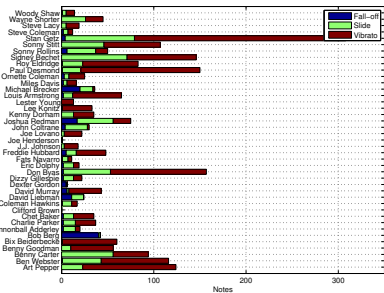


Figure 2: Number of tones of each artist which are annotated with fall-off, slide, and vibrato.

4.2 Score-informed Source Separation

To separate the solo/lead instrument from the backing track, the method for pitch-informed solo and accompaniment separation proposed in [4] was used. For this study, the automatic pitch detection stage in the separation algorithm was bypassed, and the manual melody transcriptions were used as prior information. The separation method is based on an iterative modeling of the solo instrument in the spectral domain. The model is constructed taking into account characteristics of musical instruments such as common amplitude modulation, inharmonicity, and enforcing magnitude and frequency smoothness constraints in the estimation. The separation method has proven to be robust in the extraction of a great variety of solo instruments, as well as being particularly efficient, with computation times that allow real-time processing. The complete dataset was processed and independent signals for the solo instruments and the backing tracks were extracted.

4.3 Score-informed f_0 tracking

The original audio recordings are processed at a sampling rate of 22.05 kHz. In order to track the f_0 contour of each tone, the signal is analyzed between the annotated note onset and offset time, for which a *reassigned magnitude spectrogram* $M \in \mathbb{R}_+^{K \times N}$ with K frequency bins and N frames

Performer	Inst.	N_S	N_N	Performer	Inst.	N_S	N_N	Performer	Inst.	N_S	N_N	Performer	Inst.	N_S	N_N
Art Pepper	cl/as	4	2134	David Liebman	ss/ts	4	3286	John Coltrane	ts/ss	11	8969	Sidney Bechet	ss	2	489
Ben Webster	ts	4	1497	David Murray	ts	4	2295	Joshua Redman	ts	5	2429	Sonny Rollins	ts	10	4639
Benny Carter	as	3	1153	Dexter Gordon	tp	4	3056	Kenny Dorham	tp	6	2149	Sonny Stitt	ts	4	1284
Benny Goodman	cl	7	1154	Dizzy Gillespie	ts	4	967	Lee Konitz	as	4	1839	Stan Getz	ts	6	3253
Bix Beiderbecke	tp	4	519	Don Byas	ts	7	2022	Lester Young	ts	4	887	Steve Coleman	as	3	1353
Bob Berg	ts	5	3275	Eric Dolphy	as	2	1109	Louis Armstrong	tp	4	634	Steve Lacy	ss	4	1437
Cannonball Adderley	as	5	2623	Fats Navarro	tp	4	937	Michael Brecker	ts	4	2605	Wayne Shorter	ts	9	3013
Charlie Parker	as	6	1688	Freddie Hubbard	tp	6	2266	Miles Davis	tp	7	2080	Woody Shaw	tp	5	1822
Chet Baker	tp	6	1100	J.J. Johnson	tb	2	754	Ornette Coleman	as	3	1782				
Clifford Brown	tp	4	1676	Joe Henderson	ts	6	3830	Paul Desmond	as	8	2142				
Coleman Hawkins	ts	6	2613	Joe Lovano	ts/ts-c	2	1787	Roy Eldridge	tp	6	1744				

Table 1: Overview over all artists in the dataset. For each artist, the number of solos N_S , the total number of notes N_N , as well as the instrument is given (ts: tenor saxophone, ss: soprano saxophone, as: alto saxophone, cl: clarinet, tp: trumpet, cor: cornet, tb: trombone, ts-c: C melody tenor saxophone).

is computed. We use a logarithmic frequency axis with a high resolution of 50 bins/semitone and a frequency range of ± 2 semitones around the annotated pitch. Based on an initial short-time Fourier transform (STFT) with a block-size of 1024, a hopsize of 64, and a zero-padding factor of 16, the magnitude values are mapped (reassigned) towards the frequency bins that correspond to their instantaneous frequency values at the original frequency bins computed using the method proposed by Abe in [1]. Two steps are performed for each tone to estimate its f_0 contour. First, we estimate a suitable *starting frame* within the tone’s duration with a prominent peak close to the annotated pitch. Second, we perform a *contour tracking* both forwards and backwards in time. Further details are provided in [3].

4.4 Tuning Frequency Estimation

The oldest recordings in our dataset date back to the year 1924, two years before the American music industry recommended 440 Hz for A4 as standard tuning, and 12 years before the American Standards Association officially adopted it. Hence, we can not rely on the assumption of a constant and fixed overall tuning. Moreover, the technical level of recording studios were rather low at this time, which might result in tuning deviations by speed fluctuations of recording machines as well as from instruments tuned to another reference pitch such as studio or live venue pianos. Hence, we estimate a *reference tuning frequency* f_{ref} prior to the intonation analysis of the solo instrument from the backing track of the rhythm section, which we obtain from the source separation process explained in Section 4.2. The reference tuning frequency corresponds to the fundamental frequency of the pitch A4 in the backing track.

In the Chroma Toolbox [13], a triangular filterbank is generated based on a given tuning frequency in such way that its center frequencies are aligned to the chromatic scale within the full piano pitch range. For a given audio signal, the magnitude spectrogram is averaged over the full signal duration and processed using the filterbank. By maximizing the filterbank output energy over different tun-

ing frequency hypotheses, a final tuning frequency estimate \hat{f}_{ref} is derived. We modified the originally proposed search range for \hat{f}_{ref} to $440 \text{ Hz} \pm 0.5$ semitone (corresponding MIDI pitch range: 69 ± 0.5) and the stepsize to 0.1 cents. As will be shown in Section 5.1, the influence of source separation artifacts on the estimation accuracy of the reference tuning frequency can be neglected.

4.5 Feature Extraction

Based on the estimated contour $f_0(n)$ of each tone, we first perform a smoothing using a two-element moving average filter in order to compensate for local irregularities and possible estimation errors. The extracted audio features describe the *gradient* of the f_0 contour as well as its temporal *modulation*. Table 2 lists all computed audio features and their dimensionality.

Category	Feature Label	Dim.
Gradient	Linear slope	1
Gradient	Median gradients (first half, second half, overall)	3
Gradient	Ratio of ascending frames	1
Gradient	Ratio of ascending / descending / constant segments	3
Gradient	Median gradient of longest segments	1
Gradient	Relative duration of longest segments	1
Gradient	Pitch progression	1
Modulation	Modulation frequency [Hz]	1
Modulation	Modulation dominance	1
Modulation	Modulation range [cent]	1
Modulation	Number of modulation periods	1
Modulation	Average relative / absolute f_0 deviation	2
Modulation	f_0 deviation inter-quartile-range	1

Table 2: Summary of audio features to describe the f_0 contours.

4.5.1 Gradient features

Based on the gradient $\Delta f_0(n) = f_0(n+1) - f_0(n)$, we first determine frames and segments of adjacent frames with ascending ($\Delta f_0(n) > 0$), descending ($\Delta f_0(n) < 0$), and constant frequency. We use the relative duration (with respect to the note duration) of each segment class as fea-

tures. Also, we compute median gradients in the first and second halves, over the whole note, as well as over the longest segment. Overall pitch progression is measured by the difference of average f_0 values in the end and beginning of each tone. Furthermore, we use linear regression to estimate the linear slope of the f_0 contour.

4.5.2 Modulation features

We analyze the modulation of the f_0 contour by computing the autocorrelation over $f_0(n)$. Fletcher [7] reported for woodwind instruments that a vibrato frequency range between 5 and 8 Hz is comfortable for listeners and common for players. We add a safety margin of 2 Hz and search for the lag position τ_{\max} of the highest local maximum within the lag range that corresponds to fundamental frequency values of $f_{\text{mod}} \in [3, 10]$ Hz and estimate the modulation frequency as $\hat{f}_{\text{mod}} = 1/\tau_{\max}$. The difference between the maximum and median magnitude within this frequency band is used as dominance measure for the modulation. Other applied features are the number of modulation periods and the frequency modulation range in cent.

4.6 Analysis of Intonation and Modulation Techniques

We distinguish three modulation techniques *fall-off*, *slide*, and *vibrato*. Table 3 provides a description of the characteristic f_0 contour shape for each technique. The number of tones in our dataset annotated with each technique is given in Table 3.

Technique	Description	Notes
Fall-off	Drop of the f_0 contour in the end of the tone after a stationary part.	146
Slide	Rise or drop of the f_0 in the beginning of the tone towards a stationary part.	708
Vibrato	Periodic modulation of the f_0 contour during the stationary part of the tone.	1380
None	No discernible modulation of the f_0 contour / No modulation technique annotated.	83587

Table 3: Frequency modulation techniques considered and number of annotated notes in the dataset for each technique.

5. RESULTS

5.1 Influence of Source Separation on the Reference Tuning Estimation

After the application of source separation algorithms, parts of the isolated solo instrument often remain audible in the backing track due to artifacts or interference. We first investigated the influence of the source separation step described in Section 4.2 on the reference tuning estimation. A subset of 13 solos was randomly selected from the dataset, covering various recording decades and solo instruments. For each solo, we took a 20s segment from the original recording, where only the rhythm section and no solo instrument is playing. We used the tuning estimation method described in Section 4.4 on both this 20s segment as well

as on the backing track obtained from the source separation of the solo part (compare Section 4.2) to get two estimates $f_{\text{ref}}^{\text{NoSolo}}$ and $f_{\text{ref}}^{\text{Backing}}$ of the reference tuning frequency.

The results show a very high sample correlation of $r = 0.97$ ($p < 0.001$) and a small root mean squared error of $\text{RMSE} = 1.05$ Hz between both estimates. These results indicate that the influence of source separation artifacts is negligible for the tuning estimation process. Therefore, we will use $\hat{f}_{\text{ref}} = f_{\text{ref}}^{\text{Backing}}$ as an estimate of the reference tuning frequency throughout the paper.

5.2 Relationship between the Reference Tuning and the Recording Year / Decade

How did the tuning frequency f_{ref} of commercial jazz recordings change during the 20th century? Figure 3 shows the distribution of solos in the dataset over the from the 1920s to the 2000s. Moreover, the inserted boxplots illustrate the deviation $\Delta f = 1200 \log_2 \frac{f_{\text{ref}}}{440}$ between the tuning frequency f_{ref} and 440 Hz in cent.

Absolute tuning deviation $|\Delta f|$ and recording year of each solo are weakly negatively correlated ($r = -0.33$, $p < 0.001$). Hence, the absolute deviation from the tuning frequency from 440 Hz decreased over the course of the 20th century, reflecting the spread of the 440 Hz standard (1955 adopted by the International Standards Organization), as well as the progress of studio technology.

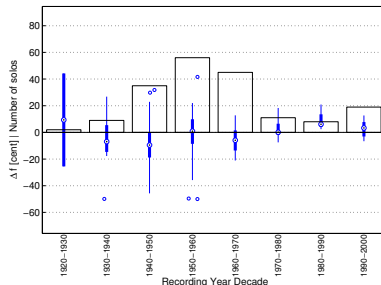


Figure 3: The box plot shows the reference tuning deviation from 440 Hz in cent for different recording year decades. The bars show the number of solos in the dataset for each decade.

5.3 Dependency of Intonation from Artist and Instrument

The distribution of the absolute deviation of tone-wise fundamental frequency values from the estimated tuning frequency as well as modulation range are shown for all musicians in Figure 4, and for all instruments in Figure 5.

According to Figure 4, the overall pitch intonation of jazz musicians is astonishingly accurate. Some woodwind and brass players tend to play a bit sharp, few a bit flat—but throughout in a range of less than 25 cent. There are

few exceptions: Sidney Bechet, a traditional soprano saxophonist, has very high values; however, presumably this is caused not by a sharp intonation but by the high percentage of pitch slides played by him (almost 15 % of the tones, cf. Figure 2).

For most players, the range of frequency modulation, i.e., the size of vibrato, is around 25 cent. There are some bigger modulation ranges from 35 to 50 cent, predominantly used by tenor saxophone players associated with swing style (Ben Webster, Coleman Hawkins, Don Byas, and Lester Young), but also by postbop tenor saxophonist Joe Lovano, and, again, by Sidney Bechet, showing the larger variance of modulation ranges. Therefore, there are some slight personal and stylistic peculiarities in the use of vibrato size. However, there are no obvious trends of intonation according to different instruments (cf. Figure 5), since for each instrument there seem to be players who play a bit sharp as well as players who play a bit low; note that for trombone and c-melody sax there is only one musician (J.J. Johnson resp. Joe Lovano) in our sample. Likewise, there is no evidence for general trends of modulation ranges with respect to instrument.

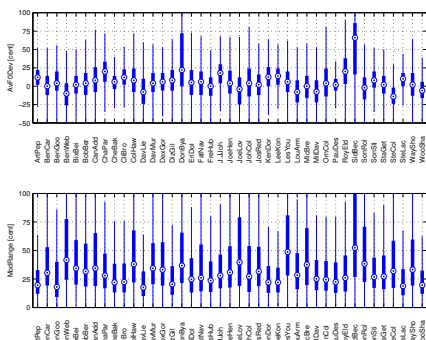


Figure 4: Absolute deviation of tone-wise fundamental frequency values from the estimated tuning frequency in cent and modulation range in cent for all musicians (for their full names see Table 1).

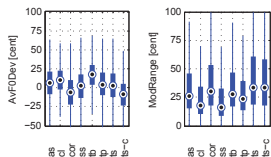


Figure 5: Absolute deviation of note-wise fundamental frequency values from the estimated tuning frequency in cent and modulation range in cent for all instruments.

5.4 Context-dependency of the Modulation Frequency of Vibrato

Does the modulation frequency of vibrato depend on pitch, or duration of the vibrato tones, or on the tempo of the piece? For the 1380 tones with vibrato notes (cf. Table 3), we found no significant correlations between modulation frequency and pitch ($r = 0.02, p = 0.42$), duration ($r = 0.02, p = 0.5$), nor tempo ($r = 0.0, p = 0.83$). The small effect size of the correlation indicates that despite the high variety of tempo values in the dataset (mean tempo 154.52 bpm, standard deviation 68.16 bpm), the modulation frequency only slightly increases with increasing tempo.

Furthermore, we investigated, whether and how the modulation frequency of vibrato is connected to the underlying metrical structure of a solo. We computed the ratio $r = T_{mod}/T_{solo}$ between the modulation tempo and the average tempo of the solo. The modulation tempo is computed as $T_{mod} = 60f_{mod}$. Figure 6 shows the ratio r against the average tempo of the solo. There is no evidence in our data for a strategy to adapt the modulation frequency of vibrato to integer multiples of the tempo of the piece, e.g., to use a vibrato speed according to simple subdivision of the beat (e.g. eighth notes or eighth triplets). As Figure 6 shows, for medium and fast tempos (100 to 350 bpm) the vibrato frequency varies only between the beat and the 16th note level. For slower tempos, the vibrato tempo could be up to six or seven times as fast as the beat—but rarely much faster.

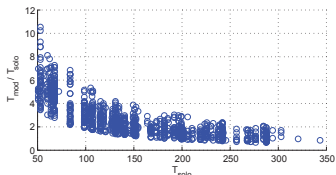


Figure 6: Ratio between the modulation frequency of vibrato tones and the average tempo of a piece vs. the average tempo.

5.5 Artist-dependency of the Modulation Frequency of Vibrato

Although there is no obvious correlation between modulation frequency and pitch, duration, or tempo, there are some peculiarities of musicians according to vibrato modulation speed. In Figure 7 only those musicians are included for which more than twenty annotated vibrato tones could be found in our data set. All in all, there seems to be no clear correlation between vibrato speed and jazz style or instrument, which indicates that modulation technique is mostly an idiosyncratic part of personal styles. Strikingly, several trumpet players can be found there (Louis Armstrong, Kenny Dorham, Roy Eldridge) using vibrato to an considerable amount and size. This is in sharp contrast to

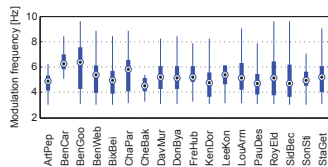


Figure 7: Modulation frequency in Hz in vibrato notes for different performers. Only performers with more than 20 vibrato notes are shown.

playing standards for brass instruments in classical music, where it is custom to play without any vibrato [7].

5.6 Automatic Classification of Frequency Modulation Techniques

Using the set of features discussed in Section 4.5, we extracted an 18-dimensional feature vector for each tone, which was used to automatically classify tones with respect to their modulation class. To this end, we only considered tones annotated with fall-off, slide, and vibrato since all remaining tones were not explicitly annotated. We used a Support Vector Machine (SVM) classifier with a linear kernel function as classification algorithm and perform a 10-fold cross-validation. Due to the imbalanced class sizes (cf. Table 3), we repeatedly re-sampled from the existing class items such that all classes have the same number of items as the largest class from the original dataset.

The confusion matrix is shown in Table 4. The highest accuracy of 92.25 % was achieved for vibrato tones. The classes fall-off and slide show lower accuracy values of 48.04 % and 67.32 %, respectively. One might assume, that the similar f_0 contour shapes of fall-offs and the slide-downs causes part of the confusions between both classes.

Correct	Classified		
	Fall-off	Slide	Vibrato
Fall-off	48.04	37.46	14.49
Slide	23.55	67.32	9.13
Vibrato	4.06	3.7	92.25

Table 4: Confusion matrix for the automatic classification of frequency modulation techniques. All values are given in percent.

6. CONCLUSIONS

In this exploratory study, we proposed a score-informed algorithm for the extraction of non-syntactical features in jazz solos played with wind and brass instruments. This method allows for an analysis of performative and expressive aspects of jazz improvisation which are not captured by the traditional approaches of jazz research such as transcriptions (even though some rudimentary notation for f_0 -modulations are used sometimes).

Combining transcriptions with state-of-art MIR algorithms significantly enhances the methodical and analytical tool box of jazz research (as well as other subfields of musicology and performance studies). In turn, this kind of fine-structured analysis might be useful in guiding automatic transcription algorithms by providing relevant background information on tone characteristics. Moreover, in this study we demonstrated exemplarily that our method can be readily applied for a range of different research questions, from historical analysis of reference tuning in 20th century jazz recordings to more general questions such as intonation accuracy or differences in f_0 modulations with respect to tempo, instrument class, stylistic trends, or personal style.

As a case study, we investigated whether some these expressive aspects, i.e., intonation, slides, vibrato speed and vibrato range, are correlated with structural features of the solos (absolute pitch, tone duration, overall tempo, meter) and whether those aspects are characteristic for an instrument, a jazz style or the personal style of a musician. While there is little evidence for a general correlation between intonation and pitch modulation (slide, vibrato) on the one hand, and structural features on the other hand, the issue of how intonation and pitch modulation contributes to the formation of a jazz style and personal style needs further examination with more data and including listening tests for style discrimination.

For the future, we plan to complete and refine the f_0 -modulation annotations for the dataset, with the overall goal of the design of an automated f_0 -modulation annotation algorithm. Finally, we aim at a complete description of personal timbre characteristics, the so-called “sound” of a player, which is an important dimension of jazz music, and not yet fully addressed. Dynamics [2], intonation, articulation, and f_0 -modulation are part of this “sound”, but other aspects such as breathiness, roughness and general spectral characteristics (and their classification) are still to be explored.

7. ACKNOWLEDGEMENTS

The Jazzomat research project is supported by a grant DFG-PF 669/7-1 (“Melodisch-rhythmische Gestaltung von Jazz-improvisationen. Rechnerbasierte Musikanalyse einstimmiger Jazzsoli”) by the Deutsche Forschungsgemeinschaft (DFG). The authors would like to thank all jazz and musicology students participating in the transcription and annotation process.

8. REFERENCES

- [1] Toshihiko Abe, Takao Kobayashi, and Satoshi Imai. Harmonics tracking and pitch extraction based on instantaneous frequency. In *Proceedings of the 1995 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 756–759, Detroit, USA, 1995.
- [2] Jakob Abeßer, Estefanía Cano, Klaus Frieler, and Martin Pfeiderer. Dynamics in jazz improvisation - score-informed estimation and contextual analysis of tone intensities in trumpet and saxophone solos. In *Proceedings of the 9th Conference on Interdisciplinary Musicology (CIM)*, Berlin, Germany, 2014.
- [3] Jakob Abeßer, Martin Pfeiderer, Klaus Frieler, and Wolf-Georg Zaddach. Score-informed tracking and contextual analysis of fundamental frequency contours in trumpet and saxophone jazz solos. In *Proceedings of the 17th International Conference on Digital Audio Effects (DAFx-14)*, Erlangen, Germany, 2014.
- [4] Estefanía Cano, Gerald Schuller, and Christian Dittmar. Pitch-informed solo and accompaniment separation: towards its use in music education applications. *EURASIP Journal on Advances in Signal Processing*, 23:1–19, 2014.
- [5] Simon Dixon, Dan Tidhar, and Emmanouil Benetos. The temperament police: the truth, the ground truth, and nothing but the truth. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 281–286, Miami, USA, 2011.
- [6] Karin Dressler. Pitch estimation by the pair-wise evaluation of spectral peaks. In *Proceedings of the 42nd AES International Conference on Semantic Audio*, pages 1–10, Ilmenau, Germany, 2011.
- [7] N. H. Fletcher. Vibrato in music – physics and psychophysics. In *Proceedings of the International Symposium on Music Acoustics*, pages 1–4, 2010.
- [8] David Gerhard. Pitch track target deviation in natural singing. In *Proceedings of the 6th International Conference on Music Information Retrieval (ISMIR)*, pages 514–519, London, UK, 2005.
- [9] Chao-Ling Hsu and Jyh-Shing Roger Jang. Singing pitch extraction by voice vibrato/tremolo estimation and instrument partial deletion. In *Proceedings of the 11th International Society for Music Information Retrieval Conference (ISMIR)*, pages 525–530, Utrecht, Netherlands, 2010.
- [10] Christian Kehling, Jakob Abeßer, Christian Dittmar, and Gerald Schuller. Automatic tablature transcription of electric guitar recordings by estimation of score- and instrument-related parameters. In *Proceedings of the 17th International Conference on Digital Audio Effects (DAFx-14)*, Erlangen, Germany, 2014.
- [11] Alexander Lerch. On the requirement of automatic tuning frequency estimation. In *Proceedings of the 7th International Conference on Music Information Retrieval (ISMIR)*, Victoria, Canada, 2006.
- [12] Sai Sumanth Miryala, Kalika Bali, Ranjita Bhagwan, and Monojit Choudhury. Automatically identifying vocal expressions for music transcription. In *Proceedings of the 14th International Society for Music Information Retrieval Conference (ISMIR)*, Curitiba, Brazil, 2013.
- [13] Meinard Müller and Sebastian Ewert. Chroma toolbox: Matlab implementations for extracting variants of chroma-based audio features. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 215–220, Miami, USA, 2011.
- [14] T. Özaslan, Serra X., and Arcos J. L. Characterization of embellishments in new performances of makam music in turkey. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, pages 13–18, 2012.
- [15] Matti P. Rynänen and Anssi Klapuri. Automatic transcription of melody, bass line, and chords in polyphonic music. *Computer Music Journal*, 32:72–86, 2008.
- [16] Justin Salamon, Geoffroy Peeters, and Axel Röbel. Statistical characterization of melodic pitch contours and its application for melody extraction. In *Proceedings of the 13th International Society for Music Information Retrieval Conference (ISMIR)*, Porto, Portugal, 2012.
- [17] Joan Serrá, Gopala K. Koduri, Marius Miron, and Xavier Serra. Assessing the tuning of sung indian classical music. In *Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR)*, pages 157–162, Miami, USA, 2011.

Author Index

- Abeßer, Jakob 749, 823
Aljanaki, Anna 770
Andrade, Nazareno 190
Arthur, Claire 728
Arzt, Andreas 357, 571
Asano, Yasuhito 371
Atkinson, Quentin D. 162
Avila Garcez, Artur d' 584
- Bansal, Jotthi 652
Baptiste, Wrena 756
Barbancho, Ana M. 735
Barbancho, Isabel 735
Batista, Gustavo E. A. P. A. 441
Beck, Serafina 79
Bello, Juan P. 248, 406, 500, 673
Bendich, Paul 38
Benetos, Emmanouil 701
Berchum, Marnix van 632
Biscainho, Luiz W. P. 264
Bittner, Rachel M. 500
Böck, Sebastian 72, 364, 625
Bogdanov, Dmitry 786
Boland, Daniel 134
Boulangier-Lewandowski, Nicolas 127
Bountouridis, Dimitrios 227
Burgoyne, John Ashley 227
Burn, David 79
- Cambouropoulos, Emilios 141, 323, 427
Cañadas-Quesada, F. J. 742
Cano, Estefanía 749, 823
Carabias-Orti, Julio José 448, 742
Caro Repetto, Rafael 107, 507
Celma, Oscar 31
Cemgil, Ali Taylan 197
Chen, Ning 598
Chen, Xuanli 79
Chen, Yuan-Ping 708
Cherla, Srikanth 584
Chi, Tai-Shih 316
Cho, Hye-Seung 639
Choi, Kahyun 779
Condit-Schultz, Nathaniel 728
Cont, Arshia 392
Crawford, Tim 524
Cumming, Julie 93
Cuvillier, Philippe 392
- Dai, Jiajie 420
Dannenber, Roger B. 17, 578, 816
- De Roure, David 211
Deruty, Emmanuel 722
Devaney, Johanna 728
Dittmar, Christian 271, 618
Dixon, Simon 127, 420, 462
Dmochowski, Jacek P. 538
Downie, J. Stephen 598, 779
Drakos, Andreas 632
Dreyfus, Laurence 211
Driedger, Jonathan 350
Duan, Zhiyao 469
Dubnov, Shlomo 176
Dutta, Shrey 605
Duval, Erik 632
Dzhambazov, Georgi 687
- Eghbal-zadeh, Hamid 554
Ehmann, Andreas F. 31
Ellis, Daniel P. W. 234, 295
Ellis, Robert J. 694
Embrechts, Jean-Jacques 462
Espinosa-Anke, Luis 100
Essid, Slim 500
- Fang, Jiakun 694
Faraldo, Ángel 364
Flexer, Arthur 547
Foster, Peter 462
Fox, Chris 524
Freedman, Dylan 561
Frieler, Klaus 823
Fu, Mutian 578
Fukayama, Satoru 114
- Gadermaier, Thassilo 571
Ganguli, Kaustuv Kanti 591
Gasser, Martin 571
Giraud, Mathieu 493
Gómez, Emilia 378
Gómez, Francisco 378
Gómez-Marín, Daniel 666
Gong, Rong 507
Gordon, Geoffrey 816
Goto, Masataka 86, 114
Gouyon, Fabien 31
Grachten, Maarten 571
Grahm, Jessica A. 763
Grill, Thomas 121, 531
Groult, Richard 493
Guiomard-Kagan, Nicolas 493
Gulati, Sankalp 680

- Gupta, Swapnil 385
- Haas, W. Bas de 483
- Handelman, Eliot 645
- Hanjalic, Alan 302
- Hankinson, Andrew 93
- Herrera, Perfecto 364, 666
- Holzappel, Andre 197
- Hörschläger, Florian 364
- Hovy, Eduard 524
- Hsu, Jennifer 176
- Hu, Xiao 779
- Huang, Yu-Hui 79
- Humphrey, Eric J. 248, 673
- Inskip, Charles 455
- Itoyama, Katsutoshi 86
- Jančovič, Peter 756
- Janer, Jordi 448
- Janssen, Berit 659
- Jentzsch, Anja 632
- Jiang, Yucong 612
- Jin, Rong 343
- Jordà, Sergi 666
- Jung, Cláudio Rosito 183
- Jure, Luis 264
- Kaliakatsos-Papakostas, Maximos 141, 323, 427
- Kaneshiro, Blair 538
- Kantan, Prithvi 591
- Kaye, Robert 786
- Khelif, Anis 330
- Kim, Hyoung-Gook 639
- Kim, Youngmoo E. 31
- Kirlin, Phillip B. 715
- Kitahara, Tetsuro 413
- Knees, Peter 65, 364
- Kohler, Eddie 561
- Köküer, Münevver 756
- Koops, Hendrik Vincent 483
- Kranenburg, Peter van 659
- Krebs, Florian 72, 625
- Kroher, Nadine 507
- Kruspe, Anna M. 336
- Kumar, Manoj 385
- Kuribayashi, Taku 371
- Le Goff, Mickael 364
- Lee, Chuan-Lung 399
- Lee, Feng-Yi 399
- Lee, Jin Ha 476, 779
- Lee, Jun-Yong 639
- Lee, Kyogu 148
- Lehner, Bernhard 309, 554, 618
- Lerch, Alexander 52, 257, 434
- Levé, Florence 493
- Lewis, Richard 211, 524
- Li, Bochen 469
- Li, Pei-Ching 809
- Liang, Che-Yuan 281
- Liang, Dawen 295
- Liebman, Elad 793
- Liem, Cynthia C. S. 302, 800
- Lin, Hsin-Ming 281
- Lin, Yin-Tzu 399
- Luo, Yin-Jyun 316
- Lykartsis, Athanasios 434
- MacFarlane, Andrew 24
- Makris, Dimos 323
- Manaris, Bill 288
- Marsden, Alan 517
- Matz, Daniel 749
- Mauch, Matthias 420
- McFee, Brian 248, 406
- McLean, Alex 517
- Melenhorst, Mark S. 800
- Miron, Marius 448
- Mok, Lillio 93
- Mora, Joaquín 378
- Müllensiefen, Daniel 227
- Müller, Meinard 271, 350, 618
- Murray-Smith, Roderick 134
- Murthy, Hema A. 385, 605
- Nakamura, Eita 392
- Ng, Kia 517
- Nieto, Oriol 406
- Nisula, Kirsten 728
- Nunes, Leonardo 264
- Nurmikko-Fuller, Terhi 211
- Ogihara, Mitsunori 204
- Ono, Nobutaka 392
- Oramas, Sergio 100, 378
- Osmalskyj, Julien 462
- Otsuka, Masaki 413
- Owen, Adrian M. 763
- Pachet, François 722
- Padilla, Victor 517

- Page, Kevin R. 211
Pandit, Vedhas 591
Park, Jeongsoo 148
Parra Chico, Gonzalo Alberto 632
Percival, Graham 114
Pfleiderer, Martin 271, 823
Popp, Phillip 59
Porter, Alastair 786
Prätzlich, Thomas 350, 618
Price, Rachel 476
Prockup, Matthew 31
- Raffel, Colin 234
Rao, Preeti 591
Raphael, Christopher 343, 612
Rastogi, Abhinav 591
Rindfleisch, Carolin 211
Ringwalt, Dan 17
Risk, Laura 93
Rocamora, Martín 264
Rodríguez-López, Marcelo E. 218
Rodriguez-Serrano, F. J. 742
Root, Deane L. 524
Rudolph, Günter 169
Ruiz-Reyes, N. 742
- Sagayama, Shigeki 392
Salamon, Justin 500
Savage, Patrick E. 162
Sbert, Mateu 735
Schaab, Maximilian 45
Schedl, Markus 65, 554
Schlüter, Jan 121, 531
Schmidt, Erik M. 31
Schramm, Rodrigo 183
Schreiber, Hendrik 241
Sekhar PV, Krishnaraj 605
Şentürk, Sertan 687
Serrà, Joan 680
Serra, Xavier 100, 107, 197, 385, 507, 680, 687, 786
Sethu, Vidhyasaharan 330
Sigler, Andie 645
Sigtia, Siddharth 127
Silva, Diego F. 441
Skowron, Marcin 65
Sordo, Mohamed 100, 204
Souza Nunes, Helena de 183
Souza, Vinícius M. A. 441
Srinivasamurthy, Ajay 197, 385
- Steffensen, Peter Berg 134
Sternin, Avital 763
Stober, Sebastian 763
Stone, Peter 793
Stoudenmier, Seth 288
Su, Alvin W. Y. 809
Su, Li 281, 316, 708, 809
Summers, Cameron 59
Sutcliffe, Richard 524
Szeto, Wai Man 155
- Tardón, Lorenzo J. 735
Thomas, David L. 715
Tralie, Christopher J. 38
Tran, Son N. 584
Tsougras, Costas 427
Tsukanov, Roman 786
Tutschku, Hans 561
- Vad, Beatrix 134
Vall, Andreu 65
Van Balen, Jan 227
Van Gool, Luc 79
Vatolkin, Igor 169
Veltkamp, Remco C. 227, 770
Vera-Candeas, P. 742
Vieira, Felipe 190
Vogl, Richard 364
Volk, Anja 218, 483, 659
- Wang, Cheng-i 176
Wang, Ye 694
Wang, Yun 816
Wasserman, Larry 578
Weigl, David M. 211
Weihs, Claus 169
Weiß, Christof 45
Weyde, Tillman 24, 584, 701
White, Corey N. 793
Widmer, Gerhard 72, 309, 357, 554, 571, 618, 625
Wiering, Frans 455, 770
Wild, Jon 645
Williamson, John 134
Wolff, Daniel 24
Wong, Kin Hong 155
Woolhouse, Matthew 652
Wu, Chih-Wei 257, 434
Wu, Ja-Ling 399
Wuchty, Stefan 204
- Xia, Guangyu 578, 816

Xiao, Haidong 598

Xing, Zhe 694

Yang, Yi-Hsuan 281, 316, 708, 809

Yao, Zun-Ren 399

Yoshii, Kazuyoshi 86

Yoshikawa, Masatoshi 371

Zacharakis, Asterios 141, 427

Zaddach, Wolf-Georg 823

Zhan, Minshu 295

Zhang, Shuo 107

Zhou, Xinquan 52

Zhu, Jie 598

Zhu, Yu 598