Music Outlier Detection Using Multiple Sequence Alignment and Independent Ensembles

Dimitrios Bountouridis(✉), Hendrik Vincent Koops, Frans Wiering, and Remco C. Veltkamp

Department of Information and Computing Sciences, Utrecht University, Utrecht, Netherlands
d.bountouridis@uu.nl

Abstract. The automated retrieval of related music documents, such as cover songs or folk melodies belonging to the same tune, has been an important task in the field of Music Information Retrieval (MIR). Yet outlier detection, the process of identifying those documents that deviate significantly from the norm, has remained a rather unexplored topic. Pairwise comparison of music sequences (e.g. chord transcriptions, melodies), from which outlier detection can potentially emerge, has been always in the center of MIR research but the connection has remained uninvestigated. In this paper we firstly argue that for the analysis of musical collections of sequential data, outlier detection can benefit immensely from the advantages of Multiple Sequence Alignment (MSA). We show that certain MSA-based similarity methods can better separate inliers and outliers than the typical similarity based on pairwise comparisons. Secondly, aiming towards an unsupervised outlier detection method that is data-driven and robust enough to be generalizable across different music datasets, we show that ensemble approaches using an entropy-based diversity measure can outperform supervised alternatives.

1 Introduction

The World Wide Web (WWW) has revolutionized music, from its creation, production, distribution to the way people currently listen to it. Due to its open nature, WWW has put users in the center of content generation. Popular websites such as *Chordify*[1], *UltimateGuitar*[2], *WhoSampled*[3] or *Midomi*[4], allow users to submit their own chord transcriptions, tabs, discovered covers or sung interpretations of a song. However, this pleasant development does not come without shortcomings; it stands to reason that user content is not always trustworthy due to human error, malicious editing and so on.

[1] www.chordify.net.
[2] www.ultimate-guitar.com.
[3] www.whosampled.com.
[4] www.midomi.com.

L. Amsaleg et al. (Eds.): SISAP 2016, LNCS 9939, pp. 286–300, 2016.
DOI: 10.1007/978-3-319-46759-7_22

Identifying and filtering out untrusted documents is of major importance for these content providing services, therefore techniques such as user ratings (e.g. star rating, social media "shares") have been frequently employed. However, ratings not only require large amount of users but, at least for chord transcriptions, have been shown to be uncorrelated to the quality of the document [18].

Outlier detection, the general task of locating those observations that *"... deviate so much from the other observations as to arouse suspicions that they were generated by a different mechanism"* [14], has found major applications in bio-informatics, fraud detection, medical diagnosis and other fields. It is therefore surprising that it has remained rather unexplored in the field of Music Information Retrieval (MIR). This can be attributed to the nature of outlier detection algorithms which by definition have two components [6]: the scores indicating level of "outliernes" of each sample, and their conversion to a binary decision by imposing thresholds based on their statistical distribution. "Outliernes" in MIR however, has been considered a welcome byproduct of music similarity. A strong music similarity model would assign low similarity scores to any documents that "do not belong." Therefore, outlier detection would be practically rendered obsolete.

In this paper we argue that, since music similarity is inherently ambiguous, it *(a)* has remained largely an unsolved problem *(b)* has been typically based on music heuristics or learning, thus becoming very task- and domain-specific and *(c)* relies on pairwise comparisons which can be time-consuming and unrealistic for large collections. Even if a strong theoretical model of similarity had been established, outlier detection in practice would be still non-trivial due to the following: first, the number of samples is usually small, and therefore an underlying "normal" model cannot be assumed or learned. Secondly, music is inherently pattern-based: therefore songs that do not belong to the reference set might share commonalities (e.g. similar chord progressions) and thus might not be deviating significantly. Consequently, the boundary between "normal" and anomalies becomes fuzzy.

Contribution: This paper's contribution is twofold. We firstly exploit the sequential nature of certain kinds of musical content, such as melodies and chord progressions, and the advantages of Multiple Sequence Alignment (MSA) in terms of sequence analysis. A sequential, music-agnostic representation of music allows for the development of tools that generalize across collections. The extensive work on MSA, mostly in the field of computational biology, allows for the adoption of tools that can separate better outlier sequences from the rest thus limiting the undesirable shortcomings of pairwise comparisons. Secondly, we present an almost settings-free outlier detection method that can find robust application to any form of music sequences. We use ensembles of different outlier solutions to form more informative decisions that avoid dependencies on specific artifacts related to a particular similarity method or data set.

Summary: The rest of the paper is organised as follows. Section 2 is a brief overview of related outlier detection approaches in MIR. Sections 3 and 4 describe

the music datasets and the basic outlier detection method considered in our work. Sections 5 and 6 break down the outlier detection into two components (similarity methods and extreme value analysis) and further analyse them. Section 7 introduces the independent ensemble approach for outlier detection, which is evaluated in Sect. 8.

2 Related Work in MIR

There are only a few published approaches that explicitly aim to tackle the task of outlier detection in music. Panteli *et al.* [20] specifically focus on world music and use data mining techniques on audio-features (e.g. rhythmic, melodic, harmonic) to detect outliers. Lukashevich and Dittmar [17], aiming towards improving a GMM mood classifier, use a Support Vector Machine (SVM) classifier as a preliminary stage to filter out outlying samples. Livshin and Rodet [16] focus on automatic removal of bad samples from an instrument music library. Their algorithms are supervised variations of the Interquartile Range. Hansen *et al.* [13] use a combination of supervised and unsupervised learning to clean-up large-scale databases that included metadata (e.g. genre information). They model the relation between metadata and audio features by training conditional densities. Unconditional densities are modeled for spotting unlikely music features. Tangential to the outlier detection problem, with common methodologies [19], is novelty detection, the automatic differentiation between known and unknown object information during testing. Most notably, Flexer *et al.* compared different rejection rules and novelty detection methods in a genre classification context [7,8]. In general, all the published approaches are either supervised and domain specific or dependent on abundant samples.

3 Music Datasets

Before going into detail about the outlier detection problem, we should describe the different music datasets investigated and how they were represented as sequences in our work (available online[5]). We use four datasets of varying size and nature, ranging from expert-annotated melodies to non-expert chord transcriptions found online. This allows us to generalize any observations, derived from this work, to almost any music dataset of sequential nature. The data sets are further explained below, while summary statistics are presented in Table 1.

The Annotated Corpus of the Meertens Tune Collections [32] is a set of 360 Dutch folk songs grouped into 26 "tune families." Each contains a group of melody variations related through an oral transmission process. For this TuneFam-26 data set, expert annotators assessed the perceived similarity of every melody over a set of dimensions (contour, rhythm, lyrics, etc.) to a set of 26 prototype "reference melodies." The Cover Song Variation data set [2], or Csv-60, on the other hand is a set of expert-annotated, symbolically-represented

[5] www.projects.science.uu.nl/COGITCH/outlier.

Table 1. Summary statistics for the four datasets of our experiments.

	TuneFam-26	Csv-60	Shs-50	Beatles
Number of cliques	26	60	50	174
Number of sequences	360	243	467	948
Avg. cliques size	13.0 *(4.0)*	4.0 *(1.1)*	9.34 *(3.5)*	5.66 *(3.61)*
Avg. sequence length	43.0 *(14.9)*	53.0 *(21.4)*	130 *(78.58)*	74.17 *(58.20)*

vocal melodies derived from matching structural segments (such as verses and choruses) of different renditions of sixty pop and rock songs. Melodies in both datasets are represented as *pitch contours*, a series of relative pitch transitions constrained to the region between +11 and −11 semitones.

The Second Hand Song dataset[6] contains metadata for around 18,000 cover songs grouped into 6,000 cliques. In order to keep computations to a practicable level, we randomly picked 50 cliques (of more than 6 songs per cliques). We denote this subset Shs-50. Songs are represented as sequences of major-minor chords by finding the shift that maximises the correlation between the Krumhansl-Kessler profiles [26] and the chroma vector of each beat-frame (as extracted by the Echonest[7] API).

From *UltimateGuitar* we web-mined 948 user chord transcriptions corresponding to the 174 songs of the complete Beatles discography as provided by the famous Beatles dataset[8]. All transcriptions in the Beatles dataset are key-normalised and reduced to major-minor chord sequences.

It should be pointed out that each group of related music sequences in the datasets is not guaranteed to contain any outliers, despite the fact that some sequences might be more dissimilar than others. In our work, the "outlier" is a randomly chosen sequence injected to the group that comes from the same dataset as the group, e.g. a chord transcription of "Let it be" injected in a group of "Yellow submarine" transcriptions by The Beatles.

4 Basic Outlier Detection

There are many outlier detection methods in the literature; however their applicability is dependent on the nature of the data, e.g. number of samples, number of outliers and so on. Therefore, picking the appropriate one, in the context of music documents, is not trivial. In this work we decided to focus on the interpretability criterion, which is of crucial importance for the analyst, since it can answer the question of why a sample is considered an outlier [6].

Extreme value analysis is the most basic and interpretable outlier detection method and has two components (see Fig. 1). It works on one-dimensional data

[6] www.secondhandsongs.com.
[7] www.echonest.com.
[8] www.isophonics.net/content/reference-annotations-beatles.

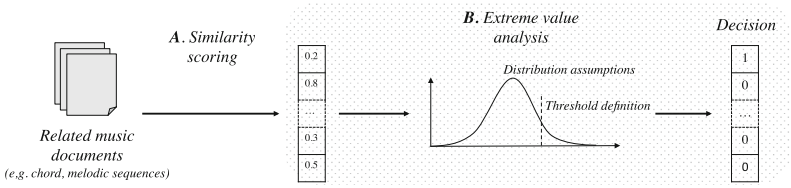

Fig. 1. The basic outlier detection pipeline using extreme value analysis.

(first component) and it assumes that outlier values are too small or too large with regard to the rest of the value distribution. In the music context, the input values correspond to the similarity $\in \mathbb{R}$ of each music document to the rest. Defining what constitutes an extreme value is typically performed by modelling the data distribution and its statistical tails (second component). The problem comes down to finding the best combination of the two components, which we analyse in the following sections. Each combination is an *outlier detection setting*.

5 Similarity Scoring Methods

We will now present the similarity scoring methods that we consider in our work (see Fig. 1, A). All are generic in the sense that they do not incorporate music heuristics. As a consequence they can be used for any form of sequences.

5.1 Pairwise Alignment

Sequence matching is typically performed using pairwise alignment (PW), the process of making two sequences have the same length by introducing gaps "-" while aligning related symbols. Sequence alignment via dynamic programming [31] is widely used for approximate string matching, and found early application in MIR. The quadratic-time Needleman and Wunsch [30] algorithm finds an optimal alignment of two sequences and returns a score that represents the cost, and can be interpreted as a quality measure. It should be noted that the score is highly dependent on the penalties for inserting and extending gaps. In our work, the PW score of a sequence in a group is the average of its pairwise scores.

In contrast to pairwise alignment, the following methods base their scoring on a multiple sequence alignment rather on multiple pairwise comparisons.

5.2 Multiple Sequence Alignment Based Methods

Before computing the similarity of each sequence in an MSA, the MSA itself needs to be computed. The optimal MSA of a group of sequences has exponential-time complexity, therefore it cannot be used in practice. Instead, the focus is on heuristic approaches that give good alignments not guaranteed to be optimal. Our work considers two algorithms: progressive alignment (PA) and MAFFT.

The most popular approach, PA, starts by building a pairwise similarity tree. Working from the leaves of the tree to the root, PA aligns the alignments, until reaching the root of the tree, where a single MSA is built. MAFFT [25] on the other hand, uses the fast Fourier transform to identify short subregions of one sequence or intermediate alignment that are high-scoring matches with same-length subregions from another sequence or alignment. Based on the MSA, our work considers the following similarity scoring methods:

Percentage Identity (PID)**.** A popular similarity scoring between pairs of sequences is the Percentage Identity (PID). It corresponds to the number of identities, meaning the number of same characters divided by the number of characters compared (gap positions excluded). In our case, the PID of a sequence in an MSA is the average of the its pairwise PID scores.

Neighbor-Joining Tree (NJ-T)**.** Neighbor-Joining Tree, is a clustering method for building phylogenetic trees of biological sequences [22]. NJ-T takes as input a distance matrix, typically based on PID, and at each stage the two nearest nodes of the tree are chosen and joined. The process is performed recursively until all of the nodes are paired and the tree is constructed. We ommit the details of the formulation of "nearest" nodes due to lack of space. We use the branch length from each leaf (sequence) to the root as a measure of similarity.

Majority-Vote Consensus (MJ-CONC)**.** A multiple alignment can be summarized to generate a single sequence that we call *consensus*. For each column, the voting process determines if the frequency of the most common symbol is above a threshold. If so, that symbol represents that column in the consensus; if not, the column is represented by an ambiguous symbol. The similarity score of each sequence is the pairwise alignment score of itself with the consensus.

Data Fusion (DF)**.** Data Fusion can be seen as an extension of majority voting in the sense that in addition to finding the most common symbol per column, it also uses the agreement between rows as a weight to favor values of rows with higher agreement [3]. The data fusion consensus happens in two steps: after computing the probabilities for each symbol for each column, a *source accuracy* is computed for each row by taking the mean of its column probabilities. The values of each row are then weighted by multiplying them with their source accuracy. The intuition is that rows with higher agreement with other rows will be more trustworthy. The process of computing symbol probabilities and source accuracy is repeated until the probabilities of the values converge. For each row, the value with the highest probability is taken as the output value. In our work we consider the source accuracy of each sequence as its similarity to the rest.

Profile Hidden Markov Models (HMM-P)**.** Profile Hidden Markov Models (HMMs) are essentially generative probabilistic automata that represent a multiple sequence alignment as a probabilistic, position-dependent scoring system [4]. Profile HMMs contain five types of states whose details are omitted due to lack of space. A query sequence can be represented as a stochastic traversal of the profile's states. As a consequence, comparing an arbitrary sequence to a

profile HMM is performed by using the Viterbi or the Forward algorithm [5] for Markov models. The similarity of each sequence is the output of its comparison to a profile HMM built from the remaining sequences.

Alignment Gap Metric (Gap-based)**.** Jehl *et al.* [15] presented an MSA-based outlier detection pipeline (called `OD-seq`) that majorly focused on efficiency since it was aimed to be used on large number of biological sequences. As such it used a gap-based distance metric that considered gap-less pairwise alignments to be of high quality and vice versa. Variations that distinguished between fewer, longer gaps or more, shorter gaps were also presented. In our work, we consider only the linear metric that treats all gaps equally.

5.3 Inlier-Outlier Separation

Although the similarity scoring methods are merely components of the whole outlier detection pipeline, we are interested in answering to which extent each similarity can better separate between outlier and inlier music sequences. Optimally a scoring method should assign high similarity scores to inliers and low scores to outliers with as minimal overlap as possible.

For each similarity method (besides PW) we compute the two score distributions (outlier and inlier sequences) over each dataset and MSA method. PW is computed only for each set since it is independent of the MSA. However, since pairwise alignment is dependent on the gap penalties, we try three different gap open and gap extend settings (.5, .2), (.8, .4) and (1, 0) with an identity substitution matrix (matches get a score of 1, while mismatches a score of −1). Table 2 presents the area under the receiver operating characteristic (ROC) curve, typically called AUC, for each scoring over each dataset and MSA algorithm.

We can make the following observations: first, there is at least one similarity method for each dataset that has higher AUC score than pairwise alignment.

Table 2. Area Under the Curve (AUC) % for each similarity method over each dataset and MSA type.

	Csv-60		TuneFam-26		Shs-50		Beatles	
	MAFFT	PA	MAFFT	PA	MAFFT	PA	MAFFT	PA
PW-.5-.2	95.95	95.95	96.52	96.52	61.66	61.66	86.34	86.34
PW-.8-.4	96.44	96.44	96.66	96.66	62.64	62.64	86.86	86.86
PW-1.0-.0	97.16	97.16	96.73	96.73	61.35	61.35	91.18	91.18
Pid	97.47	91.69	91.82	79.1	60.55	54.38	87.96	88.06
NJ-T	98.5	95.07	89.56	82.09	60.55	55.96	84.9	85.34
MJ-conc	99.07	95.09	91.68	88.81	57.7	54.42	83.74	80.39
DF	**99.22**	95.65	92.12	83.14	63.79	57.52	87.97	88.25
HMM-P	94.88	91.11	**98.29**	96.84	**75.62**	66.7	**94.76**	93.25
Gap-based	91.65	89.46	92.99	90.55	65.55	58.19	83.5	88.17

Therefore, we have shown that one can use MSA as a more reliable basis for outlier detection. Secondly, profile HMMs show generally the highest separation between outliers and inliers over all datasets. Thirdly, it becomes obvious that the MAFFT-based MSA results to generally higher separation than progressive alignment. Finally, Csv-60 and Shs-50 seem to be the "easiest" and most difficult datasets respectively to perform outlier detection on.

6 Extreme Value Analysis Algorithms

In the previous section we presented a number of similarity scoring methods. In this section we briefly present five extreme value analysis algorithms that work on top of the similarity scores (see Fig. 1, B). The list is definitely incomplete, however it captures a wide range of algorithms.

Thresholding, classifying as outliers those documents with similarity smaller than a predefined threshold θ, is the simplest form of extreme-value outlier detection. The **Z-score** is one of the simplest ways to avoid using a fixed absolute threshold. It represents the amount of standard deviations σ a value is from the mean. However, one should decide on a threshold θ_z, above which a value would be considered anomalous. The **Grubb's test** [11] is used to detect single outliers on data following approximately a normal distribution. The Grubb's test statistic is the largest absolute deviation from the sample mean in units of the sample standard deviation. Grubb's test requires us to decide on the significance level a. Since the presence of outliers is likely to affect the mean and the standard deviation, Z-score and Grubb's test can become unreliable. The median, however, is typically more robust to outliers. The **Median Absolute Deviation (MAD)** score is the median of the absolute deviations from the data's median. A data point x_i is considered an outlier if $|x_i - median(X)|/MAD$ is larger than θ_{MAD}. Finally, the **percentile** is used in statistics to indicate the value below which a given percentage of observations fall. For large normally-distributed populations, percentiles represent the area under the normal curve, increasing from left to right. A percentile based outlier detection method requires us to set a percentile threshold θ_p above which a value is considered anomalous.

6.1 Evaluation

In order to evaluate each extreme value analysis algorithm, one should analyse their behaviour with respect to their corresponding threshold and each similarity scores they were applied on. However, we are rather interested in validating our initial hypothesis; similar to any classification problem, it comes to reason that different *outlier detection settings* (similarity scoring and outlier algorithm) would behave differently depending on the nature of the dataset.

For each of our music datasets, we brute-force find a single *outlier detection setting* that results to the best overall outlier prediction (measured with the $F1$ score). We then apply the same setting on the remaining datasets. The $F1$ scores

Table 3. The $F1$ score for each optimised setting applied to all datasets. The standard deviation (*std*) for each dataset is also presented. The optimised settings per dataset are: CSV-60$_{setting}$: NJ-T & z-score & $\theta_z = 1.56$, TUNEFAM-26$_{setting}$: HMM-P & z-score & $\theta_z = 1.8$, SHS-50$_{setting}$: HMM-P & thresholding & $\theta = .22$, BEATLES$_{setting}$: HMM-P & thresholding & $\theta = .13$.

	CSV-60	TUNEFAM-26	SHS-50	BEATLES
CSV-60$_{setting}$	**0.945**	0.676	0.588	0.725
TUNEFAM-26$_{setting}$	0.639	0.**889**	0.630	0.610
SHS-50$_{setting}$	0.665	0.820	0.**668**	0.683
BEATLES$_{setting}$	0.501	0.684	0.626	**0.769**
std	0.161	0.090	0.028	0.058

for each setting applied to each dataset are presented in Table 3. It becomes obvious that different optimised settings applied to different datasets result in major fluctuations in performance.

7 Independent Ensembles

In contrast to the previously presented methods, we are interested in an unsupervised outlier detection method that is parameter-free and generalizable so that it can be applied almost "out-of-the-box" in various music collections.

Independent ensembles [6] are based on different instantiations (parameter settings) of one or more outlier detection algorithms. The parameters even the algorithms themselves can be randomly selected and the output of their execution is combined to form the final decision. The principle behind independent ensembles is to achieve robustness by avoiding dependencies on specific artifacts related to a particular algorithm or data set. In addition, it is assumed more difficult to design a single sophisticated algorithm than to optimize the combination of algorithms with relatively lower complexity [9].

Independent ensembles, depending on the task and context, appear with different names in the literature ranging from "committees" and Multiple Classifier Systems (MCS) to clustering or classification ensembles. However, the fundamental problem is generally the same: given an unlabeled data set $D = \{x_1, x_2, ..., x_n\}$ and a set of classification solutions $\{C_1, C_2, ..., C_k\}$ that map the data to a class $f_j(x) = m$ we are interested in a single "resultant" solution f^* that combines the classification solutions. The "accuracy" of the ensemble is measured by the match between the solution produced and the reference ground-truth. The problem, in our outlier-detection-for-music context, can be considered a special, binary case with $m \in \{0, 1\}$ (where 0 corresponds to "inlier" and 1 to "outlier") and with the classification solutions being the output of different *outlier detection settings*.

7.1 Diversity

An ongoing issue with ensembles in general is how to select the set of classification solutions. A set of similar classification solutions is not guaranteed to lead to the best solution, therefore the concept of *diversity* has been introduced. Although diversity has been shown to be fundamental for an ensemble's success [9], a consistent relation between the ensemble's diversity and the solution's accuracy has not been shown. In addition, diversity can be formulated in various ways. Hadjitodorov *et al.* [12] have shown the potential of moderate diversity based on Adjusted Rand Index [21], but avoided generalizing their observations.

7.2 Diversity Experiment

In this section we describe an experiment to investigate the behaviour of different diversity measures on music datasets. We also answer whether Hadjitodorov's hypothesis, denoted as Soft-Correlation Rule by [9], holds for our particular task.

Each dataset is split into a *training* and a *test set* (70 %–30 % split). The training sets combined form the *grouped training set*. For each group of related music documents in the *grouped training set*, we compute the output-solution of every possible *outlier detection setting*. We shuffle them and randomly select 10 ensembles of 25 solutions each, similar to [12]. We finally compute the diversity and quality of each ensemble. There are three important factors to consider.

As Zimek [23] mentions, an important issue with outlier detection ensembles is how to combine the different individual solutions to derive a consensus or ensemble result. The problem is not trivial especially when the outlier detectors output solutions that are ranked lists or score vectors. In our case however, each solution is a binary vector we therefore employ the intuitive majority voting.

The second important issue is the question of how to measure an ensemble's quality. Given a ground truth it is typical to use measures such as Rand index and Adjusted Rand index [23]. Similarly one can compute the average F-measure score over all solutions as we do in our paper. Yet using ground truth information to assess quality is debatable [22], especially considering we are aiming for an unsupervised outlier detection method, however as [23] states "there is probably no better approach available to assess clustering quality w.r.t. external knowledge." The third issue is which diversity measures should be investigated. Our work considers all diversity measures described [12] which are divided into two general groups, pairwise and non-pairwise. The pairwise method D_P, is based on the Adjusted Rand Index and on pairwise comparisons between all the solutions in the ensemble. Four of the non-pairwise measures D_{np-1},D_{np-2}, D_{np-3} and D_{np-4} are based on the difference of each solution from the final ensemble decision. The measure proposed by [10], denoted by [12] as H, is based on the entropy of the "consensus" matrix, which stores the co-agreement between all ensembles solutions. In our work, we consider two additional diversity measures, denoted Df_{sa} and E, which are based on data fusion and entropy respectively. In Sect. 5.2 we presented data fusion, a byproduct of which is a *source accuracy* measure for every sequence in the set that we aim to "fuse." Considering that

high source accuracy corresponds to all sequences being similar and vice versa, one can use their average complement as a measure of diversity. The E measure makes use of the binary nature of our task which allows us to convert each solution vector to a decimal representation, thus represent the ensemble as a sequence of decimal numbers. The entropy of the sequence can act as a measure of diversity.

Scatter plots of the different diversity measures plotted against the ensemble quality are presented in Fig. 2 (four omitted due to lack of space). The Pearson and Spearman's coefficients for the linearity and monotonicity tests respectively are presented in Table 4. It becomes obvious that the entropy-based diversity measure E is the most correlated to the ensemble quality. Df_{sa} follows (negative correlation), while the rest do not show any particular pattern of correlation (e.g. the closer to the median the better). Therefore, the Soft-Correlation rule cannot be applied to our particular case. Based on the results we can hypothesize that the best strategy for music outlier detection using independent ensembles is picking the ensemble with the highest diversity as measured by the measure E.

Table 4. The Pearson's and Spearman's coefficients for linearity and monotonicity tests.

	D_P	D_{np-1}	D_{np-2}	D_{np-3}	D_{np-4}	H	E	Df_{sa}
Pearson's	−0.12	−0.13	0.08	0.10	0.09	0.05	0.48	−0.23
Spearman's	−0.11	−0.11	0.02	0.06	0.09	0.03	0.51	−0.21

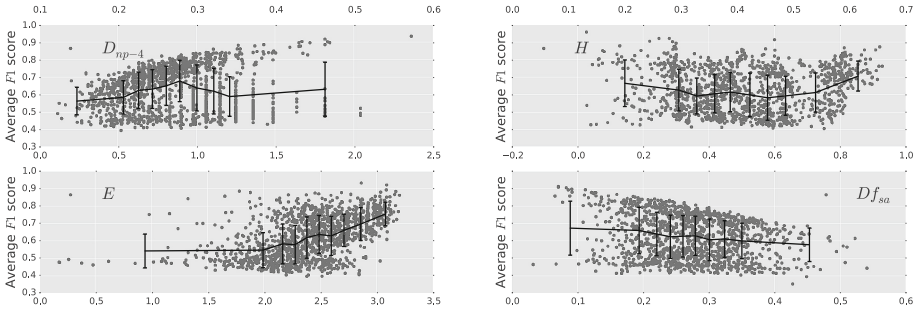

Fig. 2. Diversity (x-axis) versus quality (measured as average $F1$ score) scatter plots for four of the eight different diversity measures.

8 Independent Ensembles Evaluation

We are interested in evaluating our independent ensembles method with regard to its outlier detection generalization ability. We compare it against two supervised approaches that do not treat all datasets independently: (a) a system that

applies the same, but optimised, outlier setting to all datasets, and (b) a system that applies a setting to each dataset i, optimised for a different dataset j.

The first supervised approach learns the *outlier detection setting* that maximizes the prediction ability (measured using the $F1$ score) on the *grouped training set*. This model, denoted *Full*, is therefore trained on all music datasets but aims at finding the setting that balances among them. The learned setting is applied on each music dataset in the *test set*. The second supervised method, called *Individual*, firstly learns the optimal *outlier detection settings* for each music dataset in the *training set*. Secondly, for each example in a music dataset belonging to *test set* it randomly applies a learned setting from a different set.

For each example in the *test set*, our ensemble approach that uses an entropy-based diversity (called *Ensemble-E*) starts by computing the output-solution of every possible *outlier detection setting*. We shuffle them and randomly select 50 ensembles of 25 solutions each. We finally select the ensemble with the highest diversity E and compute the final prediction.

The $F1$ scores of a 40-fold, cross-validation are presented in Table 5. We compare the methods' distributions using the Wilcoxon rank-sum test (the scores do not follow a normal distribution). The results show that the *Ensemble-E* method significantly outperforms the rest for all datasets not only as a whole (see "Grouped" on Table 5) but individually also. For Csv-60 the *Full* method shows better performance than *Ensemble-E* but not at a significant level.

Table 5. $F1$ scores for the three outlier detection approaches. $p_{i,e}$ and $p_{f,e}$ are the p-value for the statistical significance tests between *Ensemble-E* vs *Individual* and *Ensemble-E* vs *Full* respectively.

	Individual	*Full*	*Ensemble-E*	$p_{i,e}$	$p_{f,e}$
Csv-60	0.61 *(0.09)*	**0.92** *(0.04)*	0.91 *(0.02)*	$< 10^{-7}$	0.0526
TuneFam-26	0.72 *(0.07)*	0.79 *(0.05)*	**0.81** *(0.04)*	$< 10^{-7}$	0.0141
Shs-50	0.56 *(0.04)*	0.55 *(0.03)*	**0.58** *(0.04)*	0.0187	$< 10^{-5}$
Beatles	0.68 *(0.04)*	0.74 *(0.03)*	**0.77** *(0.02)*	$< 10^{-7}$	$< 10^{-7}$
Grouped	0.67 *(0.02)*	0.75 *(0.02)*	**0.77** *(0.02)*	$< 10^{-7}$	$< 10^{-7}$

Our experiments so far were based on the assumption that only one outlier can exist in a group of related music documents. In reality the outliers can be more (but definitely less than half the number of documents). Table 6 presents the results for the same experiment applied on the same train and test sets but now with two outliers per group. Our ensemble approach shows again superior performance for all sets individually (except for Csv-60) and as a whole.

9 Discussion and Conclusions

Working towards an unsupervised outlier detector for music sequences we presented an ensemble approach that outperformed supervised approaches. However,

Table 6. $F1$ scores for the three outlier detection approaches on the datasets with two outliers per group.

	Individual	*Full*	*Ensemble-E*	$p_{i,e}$	$p_{f,e}$
Csv-60	0.73 *(0.05)*	0.80 *(0.03)*	**0.81** *(0.03)*	$< 10^{-7}$	0.5908
TuneFam-26	0.68 *(0.05)*	0.81 *(0.04)*	**0.84** *(0.04)*	$< 10^{-7}$	$< 10^{-4}$
Shs-50	0.55 *(0.03)*	0.53 *(0.04)*	**0.58** *(0.03)*	$< 10^{-5}$	$< 10^{-7}$
Beatles	0.65 *(0.04)*	0.75 *(0.04)*	**0.77** *(0.02)*	$< 10^{-7}$	$< 10^{-2}$
Grouped	0.66 *(0.02)*	0.74 *(0.03)*	**0.76** *(0.01)*	$< 10^{-7}$	$< 10^{-2}$

we should be careful before generalizing our observations. The diversity measure employed was selected based on ground truth knowledge. And although there is currently no other way to assess the relationship between a diversity measure and quality, we should avoid calling our approach strictly "unsupervised." In addition, the effect of the ensemble size (beside the number of ensembles from which we pick one) was not investigated and should be addressed in future work.

Despite these reservations our approach shows great potential due to the following: it is based on interpretable components, namely MSA-based similarity and extreme value analysis. MSA is a structure that potentially holds information that is left unexplored using pairwise comparisons. Extreme value analysis is intuitive and extremely efficient. Combining outlier detectors using ensembles, renders the approach domain-agnostic. This advantage is not be taken lightly: modelling the domain is fundamental for any outlier detection algorithm [6], therefore avoiding it for an unknown music dataset can be extremely beneficial.

References

1. Bertin-Mahieux, T., Ellis, D.P., Whitman, B., Lamere, P.: The million song dataset. In: Proceedings of the 12th International Society for Music Information Retrieval Conference, pp. 591–596 (2011)
2. Bountouridis, D., Van Balen, J.: The cover song variation dataset. In: The International Workshop on Folk Music Analysis (2014)
3. Dong, X.L., Berti-Equille, L., Srivastava, D.: Integrating conflicting data: the role of source dependence. Proc. VLDB Endow. **2**(1), 550–561 (2009)
4. Eddy, S.R.: Profile hidden Markov models. Bioinformatics **14**(9), 755–763 (1998)
5. Eddy, S.R.: Accelerated profile HMM searches. PLoS Comput. Biol. **7**(10), e1002195 (2011)
6. Aggarwal, C.C.: Outlier analysis. In: Aggarwal, C.C. (ed.) Data Mining, pp. 237–263. Springer, New York (2015)
7. Flexer, A., Pampalk, E., Widmer, G.: Novelty detection based on spectral similarity of songs. In: ISMIR, pp. 260–263 (2005)
8. Flexer, A., Schnitzer, D.: Using mutual proximity for novelty detection in audio music similarity. In: Proceedings of 6th International Workshop on Machine Learning and Music (MML), pp. 31–34. Citeseer (2013)

9. Freitas, C.O.A., Carvalho, J.M., Oliveira, J.J., Aires, S.B.K., Sabourin, R.: Confusion matrix disagreement for multiple classifiers. In: Rueda, L., Mery, D., Kittler, J. (eds.) CIARP 2007. LNCS, vol. 4756, pp. 387–396. Springer, Heidelberg (2007). doi:10.1007/978-3-540-76725-1_41
10. Greene, D., Tsymbal, A., Bolshakova, N., Cunningham, P.: Ensemble clustering in medical diagnostics. In: 17th IEEE Symposium on Computer-Based Medical Systems, CBMS 2004, Proceedings, pp. 576–581. IEEE (2004)
11. Grubbs, F.E.: Sample criteria for testing outlying observations. Ann. Math. Stat. **21**, 27–58 (1950)
12. Hadjitodorov, S.T., Kuncheva, L.I., Todorova, L.P.: Moderate diversity for better cluster ensembles. Inf. Fusion **7**(3), 264–275 (2006)
13. Hansen, L.K., L.-Schioler, T., Petersen, K.B., Arenas-Garcia, J., Larsen, J., Jensen, S.H.: Learning and clean-up in a large scale music database. In: 2007 15th European Signal Processing Conference, pp. 946–950. IEEE (2007)
14. Hawkins, D.M.: Identification of Outliers, vol. 11. Springer, Netherlands (1980)
15. Jehl, P., Sievers, F., Higgins, D.G.: OD-seq: outlier detection in multiple sequence alignments. BMC Bioinf. **16**(1), 269 (2015)
16. Livshin, A., Rodet, X.: Purging musical instrument sample databases using automatic musical instrument recognition methods. IEEE Trans. Audio Speech Lang. Process. **17**(5), 1046–1051 (2009)
17. Lukashevich, H., Dittmar, C.: Improving GMM classifiers by preliminary one-class svm outlier detection: application to automatic music mood estimation. In: Locarek-Junge, H., Weihs, C. (eds.) Classification as a Tool for Research, pp. 775–782. Springer, Heidelberg (2010)
18. Macrae, R., Dixon, S.: Guitar tab mining, analysis and ranking. In: ISMIR, pp. 453–458 (2011)
19. Markou, M., Singh, S.: Novelty detection: a reviewpart 1: statistical approaches. Signal Process. **83**(12), 2481–2497 (2003)
20. Panteli, M., Benetos, E., Dixon, S.: Automatic detection of outliers in world music collections. In: Fourth International Conference on Analytical Approaches to World Music (AAWM 2016) (2016)
21. Rand, W.M.: Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. **66**(336), 846–850 (1971)
22. Saitou, N., Nei, M.: The neighbor-joining method: a new method for reconstructing phylogenetic trees. Mol. Biol. Evol. **4**(4), 406–425 (1987)
23. Zimek, A., Campello, J.G.B., Sander, J.: Ensembles for unsupervised outlier detection: challenges and research questions a position paper. ACM SIGKDD Explor. Newsl. **15**(1), 11–22 (2014)
24. Gómez, E., Klapuri, A., Meudic, B.: Melody description and extraction in the context of music content processing. J. New Music Res. **32**(1), 23–40 (2003)
25. Katoh, K., Misawa, K., Kuma, K.-I., Miyata, T.: MAFFT: a novel method for rapid multiple sequence alignment based on fast fourier transform. Nucleic Acids Res. **30**(14), 3059–3066 (2002)
26. Krumhansl, C.L., Kessler, E.J.: Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. Psychol. Rev. **89**(4), 334 (1982)
27. Li, S.Z.: Content-based audio classification and retrieval using the nearest feature line method. Speech Audio Process. **8**(5), 619–625 (2000)
28. Malt, B.C.: An on-line investigation of prototype and exemplar strategies in classification. J. Exp. Psychol. Learn. Mem. Cogn. **15**(4), 539 (1989)

29. Martin, B., Brown, D.G., Hanna, P., Ferraro, P.: Blast for audio sequences alignment: a fast scalable cover identification. In: 13th International Society for Music Information Retrieval Conference, p. 529 (2012)
30. Needleman, S.B., Wunsch, C.D.: A general method applicable to the search for similarities in the amino acid sequence of two proteins. J. Mol. Biol. **48**(3), 443–453 (1970)
31. Sankoff, D., Kruskal, J.B.: Time warps, string edits, and macromolecules: the theory and practice of sequence comparison. Addison-Wesley Publishing Company, Reading (1983)
32. van Kranenburg, P., de Bruin, M., Grijp, L., Wiering, F.: The shs-50 tune collections. In: Shs-50 Online Reports (2014)