Software Impacts 9 (2021) 100103

Contents lists available at ScienceDirect

Software Impacts

journal homepage: www.journals.elsevier.com/software-impacts

Original software publication

Implementation of and experimental software for active selection of classification features

Thomas T. Kok [a,b,*], Georg Krempl [b], Hugo G. Schnack [c,d]

[a] IDLab, Ghent University - imec, Belgium
[b] Algorithmic Data Analysis Group, Department of Information & Computing Sciences, Utrecht University, The Netherlands
[c] Department of Psychiatry, UMCU Brain Center, University Medical Center Utrecht, Utrecht University, Utrecht, The Netherlands
[d] Department of Languages, Literature and Communication, Faculty of Humanities, Utrecht University, The Netherlands

ARTICLE INFO

Keywords:
Active learning
Active feature acquisition
Active selection of classification features
Machine learning experiment evaluation framework

ABSTRACT

In some machine learning applications, obtaining data on the most predictive features is costly, but other features are readily available. Recently, first active learning approaches for this **A**ctively **S**electing **C**lassification **F**eatures problem (*ASCF*) have been proposed. In this paper, we introduce a Python package that provides a framework for ASCF, including implementations of a supervised and an unsupervised selection approach, as well as a framework for performing experimental evaluations. This framework has been used in recent publications in the context of neuroimaging research on mental disorders, where its usefulness has been demonstrated in a simulated study design with MRI data.

Code metadata

Current code version	v1
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2021-55
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/3313284/tree/v1
Legal Code License	MIT License
Code versioning system used	git
Software code languages, tools, and services used	Python
Compilation requirements, operating environments & dependencies	https://github.com/thomastkok/active-selection-of-classification-features/blob/master/requirements.txt
If available Link to developer documentation/manual	https://github.com/thomastkok/active-selection-of-classification-features/blob/master/README.md
Support email for questions	thomas.kok@ugent.be, mail@thomastkok.com

Main text

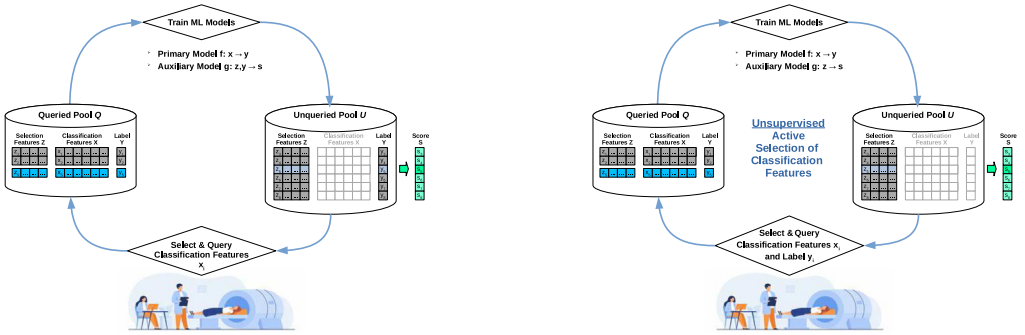
Motivation. A central prerequisite for the use of supervised machine learning techniques is the availability of data. However, in practical applications the acquisition of data is often expensive or tedious. For applications where data on predictive features are abundant, but labels are scarce and require costly acquisition from an oracle, active learning provides a rich literature on approaches for selecting the most insightful instances for labeling. However, in some applications obtaining data on the most predictive features themselves is costly, while data on other features are cheap or readily available. For these applications, the novel active learning problem of **Actively Selecting Classification Features** (ASCF) has recently been defined [1]:

Given is the primary task to learn a classifier $f : \mathrm{x} \rightarrow y$ *on predictive but expensive, yet-to-be-acquired* classification *features* x, *while another set of cheap auxiliary features* z *is available for* selection. *Then, the ASCF task consists of actively selecting these instances, for which acquiring their expensive features* x *is most useful. This is done by learning an auxiliary predictor* $h : \mathrm{z} \rightarrow \mathrm{x}$ *to predict this usefulness based on auxiliary features.*

The code (and data) in this article has been certified as Reproducible by Code Ocean: (https://codeocean.com/). More information on the Reproducibility Badge Initiative is available at https://www.elsevier.com/physical-sciences-and-engineering/computer-science/journals.

* Corresponding author at: IDLab, Ghent University - imec, Belgium.
E-mail address: thomas.kok@ugent.be (T.T. Kok).

https://doi.org/10.1016/j.simpa.2021.100103
Received 17 May 2021; Received in revised form 24 June 2021; Accepted 25 June 2021



(a) Supervised ASCF: Selection h considers labels (b) Unsupervised ASCF: Selection h without labels

Fig. 1. **A**ctive **S**election of **C**lassification **F**eatures: The instances with the most insightful costly feature values x of the primary classification task $f : x \rightarrow y$ are identified by using auxiliary cheap features z and learning a predictor of their value $h : z \rightarrow x$ and estimating their usefulness s.

For this task, shown in Figs. 1(a)–1(b), we provide a software package that

1. implements the problem setting,
2. implements two ASCF-approaches, one supervised, one unsupervised,
3. and provides a framework for the efficient execution of experiments.

The motivation for exploring this problem is motivated both by theory and by a practical application setting. The specific problem setting is unexplored in literature (with the Active Feature Acquisition [2] problem setting coming closest). The problem setting was defined originally for a medical case study, where the aim was to more efficiently build a prediction model. In this case, a classification model to predict schizophrenia diagnosis is built using MRI scans. Acquisition of these scans is expensive and uncomfortable for the patients, so being able to determine whether or not a scan of a certain patient would be informative would avoid unnecessary burden for the patient and reduce costs.

Functionality. The software was implemented using the `Python` programming language [3], relying on several packages, most importantly: `scikit-learn` [4], `numpy` [5], and `pandas` [6,7]. The open source, MIT-licensed code,[1] is divided into several subfolders:

- `approaches` and `baselines`: The approaches and baselines as defined in our previous work on ASCF is implemented and shown in these subfolders.
- `base`: This subfolder contains the essential classes for maintaining the environment of an ASCF problem setting. These are: the dataset with missing classification features, the model which is to be optimized, the oracle which is able to query any missing classification features, and the sampler which allows selecting these queries.
- `experiment`: This subfolder contains the essentials for the experimental setup. It is able to run experiments from the command line, as well as generate experimental setup and evaluate experimental results.

Impact overview. The development of this software allowed the pursuit of our existing research questions: to find if there are possibilities for improving upon random selection for this problem setting. For this, we needed to perform experiments to test any developed approaches in a real setting. This software allows the pursuit of any research question relating to the problem setting of Active Selection of Classification Features, and most importantly potential approaches for this problem. The ability to run experiments with this software has changed the daily practice of its users, as well as the ability to more easily develop new approaches for this problem. When a new approach for this problem is developed, or a dataset is found, this software allows easily exploring the initial results as well as going more into depth.

The software has been used in the following publications to obtain results, with more domain-related publications expected to follow later:

1. T. Kok, Active Selection of Classification Features, Master's thesis, Utrecht University, Utrecht, The Netherlands (2020), see [8]
2. T. Kok, R. M. Brouwer, R. M. Mandl, H. G. Schnack, G. Krempl, Active selection of classification features, in: Advances in Intelligent Data Analysis XIX. IDA 2021, Vol. 12695 of LNCS, Springer, 2021, pp. 184–195. doi: http://dx.doi.org/10.1007/978-3-030-74251-5_15

With the software, these publications were able to show improvements upon simple baselines for benchmark datasets and especially when applied to real-world neuroimaging data.

As of now, the software is not widespread, but can grow with awareness of the ASCF-problem and the availability of approaches addressing it. The need for such techniques for neuroimaging, particularly in clinical settings, has been explicitly confirmed in the reviews for [1]. Thus, it allows for much potential.

The software has potential use outside of the neuroimaging domain. Any problem that follows the principles of the ASCF problem setting, can make use of the software as well as relevant approaches. This can be checked by confirming the following items:

- Some set of information is either known for all data points, or easy and cheap to retrieve. In addition, this data is not relevant for the final classification model, or cannot be included for other reasons. This information will correspond to the selection features.
- Some set of information is not yet known for all data points, and is either hard or costly to obtain. This information will correspond to the classification features.

In the medical domain, often measurement instruments have to be chosen from a large set of possible instruments, varying in availability, patient burden, cost, time consumption. Many (classification) problems in the medical domain likely fulfill the ASCF requirements, and 'easy' features may be used to select those cases whose 'hard' features are

[1] Available at: https://github.com/thomastkok/active-selection-of-classification-features.

most informative to the model. Potential applications of ASCF could possibly be found in other (bio)medical domains such as animal research and pharmaceutical research, and, beyond that, in other fields where information gathering by or from humans have is hard or where some of the measurements are hard to obtain — think of geology.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

We thank Rachel M. Brouwer, Rene M. Mandl, Hilleke E. Hulshoff Pol and Wiepke Cahn from UMCU Brain Center, and Ad Feelders from Utrecht University. Furthermore, we thank the SIG Applied Data Science at UU/UMCU for funding the research project “Using active learning to reduce the costs of population-based neuroimaging studies”.

References

[1] T. Kok, R.M. Brouwer, R.M. Mandl, H.G. Schnack, G. Krempl, Active selection of classification features, in: Advances in Intelligent Data Analysis XIX. IDA 2021. Vol. 12695, in: LNCS, Springer, 2021, pp. 184–195, http://dx.doi.org/10.1007/978-3-030-74251-5_15.

[2] M. Saar-Tsechansky, P. Melville, F. Provost, Active feature-value acquisition, Manage. Sci. 55 (4) (2009) 664–684, http://dx.doi.org/10.1287/mnsc.1080.0952.

[3] G. Van Rossum, F.L. Drake, Python 3 Reference Manual, CreateSpace, Scotts Valley, CA, 2009.

[4] 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. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[5] C.R. Harris, K.J. Millman, S.J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. van Kerkwijk, M. Brett, A. Haldane, J.F. del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T.E. Oliphant, Array programming with numpy, Nature 585 (7825) (2020) 357–362, http://dx.doi.org/10.1038/s41586-020-2649-2.

[6] The pandas development team, pandas-dev/pandas: Pandas. (Mar. 2020) http://dx.doi.org/10.5281/zenodo.3715232.

[7] M. Wes, Data structures for statistical computing in python, in: S. van der Walt, J. Millman (Eds.), Proceedings of the 9th Python in Science Conference, 2010, pp. 56–61, http://dx.doi.org/10.25080/Majora-92bf1922-00a.

[8] T. Kok, Active Selection of Classification Features, Master’s thesis, Utrecht University, Utrecht, The Netherlands, 2020.