Software Impacts 10 (2021) 100137

Contents lists available at ScienceDirect

Software Impacts

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

Original software publication

BOOMER — An algorithm for learning gradient boosted multi-label classification rules

Michael Rapp

Knowledge Engineering Group, TU Darmstadt, Hochschulstraße 10, 64289 Darmstadt, Germany

ARTICLE INFO

Keywords:
Machine learning
Multi-label classification
Gradient boosting
Rule learning

ABSTRACT

Multi-label classification is concerned with the assignment of sets of labels to individual data points. Due to its diverse real-world applications, e.g., the annotation of text documents with topics, it has become a well-established field of machine learning research. Compared to traditional classification, where classes are mutually exclusive, multi-label classification comes with interesting challenges, most prominently the requirement to take dependencies between labels into account. In this work, we present a modular and customizable implementation of BOOMER – an algorithm for learning gradient boosted multi-label classification rules – that can flexibly be adjusted to different use cases and requirements.

Code metadata

Current code version	0.6.0
Permanent link to code/repository used for this code version	https://github.com/SoftwareImpacts/SIMPAC-2021-114
Permanent link to Reproducible Capsule	https://codeocean.com/capsule/6981524/tree/v1
Legal Code License	MIT License (MIT)
Code versioning system used	git
Software code languages, tools, and services used	C++, Cython, Python 3
Compilation requirements, operating environments & dependencies	x86_64 processor, numpy, scipy, scikit-learn, liac-arff, openmp, meson, ninja
If available Link to developer documentation/manual	https://mlrl-boomer.readthedocs.io
Support email for questions	michael.rapp.ml@gmail.com

1. Introduction

The goal of multi-label classification (MLC) is to predict a subset of relevant labels out of a predefined set of available labels. Real-world applications of MLC include the assignment of keywords to text documents, the annotation of multimedia data, such as images, videos or audio recordings, as well as applications in the field of biology. For a more extensive overview, we refer to survey articles on the topic, such as the one by Gibaja and Ventura [1]. MLC is often tackled as a supervised learning problem, where a predictive model is derived from labeled training data provided to the learning algorithm. To assess the predictive performance of a multi-label classifier by comparing the set of predicted labels to the true labels, a variety of evaluation measures with different characteristics have been proposed in the literature. As these measures may conflict with each other, optimizing for one particular measure often leads to deterioration with respect to another [2]. As a consequence, a single model is usually not able to achieve optimal results with regard to all commonly used measures. This motivates the need for MLC methods that offer means to be tailored to a particular target measure at hand and are therefore flexible enough to be used for different use cases and applications.

Among the many machine learning methods available, approaches based on gradient boosting have received great attention in recent years and have been shown to achieve state-of-the-art performance when dealing with classification tasks. Moreover, the widespread use of publicly available gradient boosting algorithms, such as XGBoost [3] or LightGBM [4], shows the demand for highly efficient and scalable implementations. Unfortunately, many algorithms, including the aforementioned ones, are limited to binary and multi-class classification and cannot deal with multi-label data out-of-the-box. Nevertheless, due to its ability to tailor models to different target functions, gradient boosting appears to be an appealing approach for solving multi-label problems. In fact, several boosting-based algorithms, specifically aimed

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.

E-mail address: mrapp@ke.tu-darmstadt.de.

https://doi.org/10.1016/j.simpa.2021.100137
Received 6 September 2021; Accepted 11 September 2021

at MLC, have been proposed in the past (e.g., [5,6], or [7]). However, they are mostly restricted to the optimization of label-wise decomposable evaluation functions, which neglect dependencies between labels that may be hidden in the data, and often lack a publicly available implementation, which impedes to use them in scientific studies or commercial applications.

In the following, we introduce an open source implementation of the BOOMER [8] algorithm, which has specifically been designed to meet the requirements of MLC problems. It utilizes the gradient boosting framework to learn ensembles of multi-label classification rules. Each of the rules included in such a model applies to a pattern encountered in the training data and provides a prediction for one or several labels. Combining the predictions of multiple rules enables the algorithm to achieve high predictive accuracy. For benchmarks that compare the predictive performance of the algorithm to those of competing approaches, we refer to preliminary work, such as [8,9]. Unlike decision trees, which are most commonly used in boosting-based approaches, individual rules do not provide predictions for all possible data points. This allows to focus the training effort on those parts of the data that are more difficult to predict. In contrast to competing approaches, BOOMER allows to optimize decomposable, as well as non-decomposable, loss functions. This flexibility, together with the ability to use different types of rules, depending on whether label dependencies should be taken into account, makes it a versatile tool for dealing with different kinds of MLC tasks.

2. Technical overview

The BOOMER software package includes the following components:

(a) An implementation of the algorithmic aspects in C++. It provides a programmatic API for configuring the algorithm and relies on OpenMP [10] for the implementation of multi-threading functionality, as well as on BLAS [11] and LAPACK [12] for linear algebra computations.
(b) A Python API that integrates with the popular scikit-learn [13] machine learning framework. It uses Cython [14] to interact with the underlying C++ implementation.
(c) Additional Python modules that help to carry out experiments using tabular datasets in the Mulan [15] format[1]. Experiments can be started via a command-line API that allows to assess the quality of predictions in terms of commonly used evaluation measures, offers means for parameter tuning and can be used to write experimental results and trained models into output files.

A key functionality of the BOOMER algorithm is its ability to optimize different loss functions. The implementation presented in this paper comes with several decomposable and non-decomposable loss functions that serve as surrogates for commonly used multi-label evaluation measures. Whereas decomposable loss functions can be optimized for each label individually, non-decomposable losses require dependencies between labels to be taken into account [2]. Depending on whether the loss function is decomposable or not and unless specified by the user, the algorithm automatically decides for the most suitable type of rules to be used, as well as a strategy for the aggregation of their predictions. As argued by Loza Mencía et al. [16] and empirically testified by Rapp et al. [8], multi-label rules that predict for several labels at the same time are well suited for the optimization of non-decomposable evaluation measures, due to their ability to capture local label dependencies, whereas single-label rules are a reasonable choice when using decomposable measures. To be able to control the characteristics of the models that are produced by the BOOMER algorithm, a variety of regularization parameters are provided, including the ability to use L_2 regularization to prevent overfitting.

[1] For example, a large collection of benchmark datasets is provided at https://www.uco.es/kdis/mllresources.

To be able to deal with large datasets, BOOMER implements the following techniques that speed up training or reduce the algorithm's memory footprint.

- The training algorithm is able to exploit sparsity in the training data, if the data points supplied for training can efficiently be stored using a sparse matrix format. For example, this requirement is often met when dealing with datasets for text classification, resulting in a significant reduction of training time.
- The algorithm allows to deal natively with both, categorical and numerical features. Therefore there is no need for pre-processing techniques, such as one-hot-encoding, which increase the dimensionality of the data and induce a computational overhead.
- The true labels, which are provided as part of the training data, can be supplied in the form of a sparse matrix. As most multi-label datasets come with sparse labels, i.e., individual data points are associated with a small fraction of the available labels, this often reduces the amount of memory required for training. Accordingly, sparse matrix formats can also be used for prediction.
- Multiple CPU cores can be utilized for training and prediction. The multi-threading implementation is based on OpenMP and the means for parallelization offered by BLAS and LAPACK, respectively.

The BOOMER algorithms is implemented in a modular fashion. This enables to use different implementations for the algorithmic aspects that are involved in the induction of rules and which are outlined by Hüllermeier et al. [17]. In the following, we provide a list of the most prominent features that can optionally be used.

- Gradient-based label binning (GBLB) [18] forms groups of labels, for which a rule should predict similarly. The use of GBLB has been found to speed up training significantly when optimizing non-decomposable loss functions and may even result in an improvement of predictive accuracy.
- Different sampling methods can be used to learn from subsets of the available training data. Among others, this includes stratification methods proposed by Sechidis et al. [19].
- Similar data points can be assigned to groups using different binning methods. Similar to the histogram-based construction of gradient boosted decision trees employed by XGBOOST [3] and LightGBM [4], this may help to process datasets with a very large number of data points.
- Early stopping strategies can be used to terminate training as soon as a model cannot further be improved in terms of the target function, according to an estimate obtained from an otherwise unused fraction of the training data.

3. Impact

By making the source code of the BOOMER algorithm publicly available, we adhere to the principles of reproducible research and enable members of the scientific community to use our approach for their own work. Since its publication, it has already been used as a baseline in empirical studies [9,20] and in the future it could further serve as a basis for developing novel machine learning algorithms. Unlike existing methods, the algorithm is not restricted to the use of label-wise decomposable evaluation functions, but can also be used for the optimization of non-decomposable measures. Due to this ability, the algorithm could lay the foundation for the development of novel machine learning algorithms, specifically tailored to the family of non-decomposable evaluation measures, as well as the investigation of corresponding surrogate losses. Besides the use for scientific purposes, the choice for a permissive free software license (MIT) allows for an integration of the algorithm with proprietary software, which is an important requirement for usage in commercial applications.

In recent years, research on multi-label classification has increasingly been motivated by the need to process large amounts of data. To account for this requirement, computational efficiency has always been a major focus of our efforts, including the investigation of optimizations and approximation techniques that may help to overcome the computational demands that result from large datasets. Despite improvements that have already been achieved in this regard (cf. [18]), the optimization of non-decomposable measures remains computationally challenging. By making the source code publicly available, we hope to contribute to the solutions of these problems.

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

This work was supported by the German Research Foundation (DFG) [grant number 400845550]. Special thanks go to Eneldo Loza Mencía, Johannes Fürnkranz and Eyke Hüllermeier for insightful discussions and constructive feedback. Furthermore, we want to thank everybody who has contributed code to the project. A frequently updated list of contributors is available in the project repository.

References

[1] Eva Gibaja, Sebastián Ventura, Multi-label learning: A review of the state of the art and ongoing research, Wiley Interdiscip. Rev. Data Min. Knowl. Discov. 4 (6) (2014) 411–444.

[2] Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, Eyke Hüllermeier, On label dependence and loss minimization in multi-label classification, Mach. Learn. 88 (1–2) (2012) 5–45.

[3] Tianqi Chen, Carlos Guestrin, XGBoost: A scalable tree boosting system, in: Proc. ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016, pp. 785–794.

[4] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, Tie-Yan Liu, LightGBM: A highly efficient gradient boosting decision tree, Adv. Neural Inf. Process. Syst. 30 (2017) 3146–3154.

[5] Yonatan Amit, Ofer Dekel, Yoram Singer, A boosting algorithm for label covering in multilabel problems, in: Proc. International Conference on Artificial Intelligence and Statistics (AISTATS), 2007, pp. 27–34.

[6] Si Si, Huan Zhang, S. Sathiya Keerthi, Dhruv Mahajan, Inderjit S. Dhillon, Cho-Jui Hsieh, Gradient boosted decision trees for high dimensional sparse output, in: Proc. International Conference on Machine Learning (ICML), 2017, pp. 3182–3190.

[7] Zhendong Zhang, Cheolkon Jung, GBDT-MO: Gradient-boosted decision trees for multiple outputs, IEEE Trans. Neural Netw. Learn. Syst. (2020).

[8] Michael Rapp, Eneldo Loza Mencía, Johannes Fürnkranz, Vu-Linh Nguyen, Eyke Hüllermeier, Learning gradient boosted multi-label classification rules, in: Proc. European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD), 2020, pp. 124–140.

[9] Eyke Hüllermeier, Marcel Wever, Eneldo Loza Mencía, Johannes Fürnkranz, Michael Rapp, A flexible class of dependence-aware multi-label loss functions, 2020, arXiv preprint arXiv:2011.00792.

[10] Rohit Chandra, Leo Dagum, David Kohr, Ramesh Menon, Dror Maydan, Jeff McDonald, Parallel Programming in OpenMP, Morgan Kaufmann, 2001.

[11] L. Susan Blackford, Antoine Petitet, Roldan Pozo, Karin Remington, R. Clint Whaley, James Demmel, Jack Dongarra, Iain Duff, Sven Hammarling, Greg Henry, et al., An updated set of basic linear algebra subprograms (BLAS), ACM Trans. Math. Software 28 (2) (2002) 135–151.

[12] Edward Anderson, Zhaojun Bai, Christian Bischof, L. Susan Blackford, James Demmel, Jack Dongarra, Jeremy Du Croz, Anne Greenbaum, Sven Hammarling, Alan McKenney, et al., LAPACK Users' guide, SIAM, 1999.

[13] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al., Scikit-learn: Machine learning in Python, J. Mach. Learn. Res. 12 (2011) 2825–2830.

[14] Stefan Behnel, Robert Bradshaw, Craig Citro, Lisandro Dalcin, Dag Sverre Seljebotn, Kurt Smith, Cython: The best of both worlds, Comput. Sci. Eng. 13 (2) (2010) 31–39.

[15] Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Jozef Vilcek, Ioannis Vlahavas, Mulan: A Java library for multi-label learning, J. Mach. Learn. Res. 12 (2011) 2411–2414.

[16] Eneldo Loza Mencía, Johannes Fürnkranz, Eyke Hüllermeier, Michael Rapp, Learning interpretable rules for multi-label classification, in: Explainable and Interpretable Models in Computer Vision and Machine Learning, Springer, 2018, pp. 81–113.

[17] Eyke Hüllermeier, Johannes Fürnkranz, Eneldo Loza Mencía, Vu-Linh Nguyen, Michael Rapp, Rule-based multi-label classification: Challenges and opportunities, in: Proc. International Joint Conference on Rules and Reasoning, 2020, pp. 3–19.

[18] Michael Rapp, Eneldo Loza Mencía, Johannes Fürnkranz, Eyke Hüllermeier, Gradient-based label binning in multi-label classification, in: Proc. European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD), 2021, pp. 462–477.

[19] Konstantinos Sechidis, Grigorios Tsoumakas, Ioannis Vlahavas, On the stratification of multi-label data, in: Proc. European Conference on Machine Learning and Knowledge Discovery in Databases (ECML-PKDD), 2011, pp. 145–158.

[20] Michael Kirchhof, Lena Schmid, Christopher Reining, Michael ten Hompel, Markus Pauly, PRSL: Interpretable multi-label stacking by learning probabilistic rules, 2021, arXiv preprint arXiv:2105.13850.