Journal of Machine Learning Research 20 (2019) 1-22 Submitted 2/17; Revised 1/19; Published 1/19

scikit-multilearn: A scikit-based Python environment for performing multi-label classification

Piotr Szymański PIOTR.SZYMANSKI@{PWR.EDU.PL,ILLIMITES.EDU.PL}
Department of Computational Intelligence
Wrocław University of Science and Technology
Wrocław, Poland

illimites foundation
Wrocław, Poland

Tomasz Kajdanowicz TOMASZ.KAJDANOWICZ@PWR.EDU.PL
Department of Computational Intelligence
Wrocław University of Science and Technology
Wrocław, Poland

Editor: Alexandre Gramfort

Abstract

The scikit-multilearn is a Python library for performing multi-label classification. It is compatible with the scikit-learn and scipy ecosystems and uses sparse matrices for all internal operations; provides native Python implementations of popular multi-label classification methods alongside a novel framework for label space partitioning and division and includes modern algorithm adaptation methods, network-based label space division approaches, which extracts label dependency information and multi-label embedding classifiers. The library provides Python wrapped access to the extensive multi-label method stack from Java libraries and makes it possible to extend deep learning single-label methods for multi-label tasks. The library allows multi-label stratification and data set management. The implementation is more efficient in problem transformation than other established libraries, has good test coverage and follows PEP8. Source code and documentation can be downloaded from `http://scikit.ml` and also via `pip`. The project is BSD-licensed.

Keywords: Python, multi-label classification, label-space clustering, multi-label embedding, multi-label stratification

1. Introduction

The Python language with its machine learning library stack has grown to become one of the leading technologies of building models for the industry and developing new methods for the researchers. While the Python community boasts with the excellent culture of development, well-defined API traditions and well-performing implementations of methods from most machine learning areas it did not have a high-quality solution for multi-label classification.

In this paper we introduce scikit-multilearn, a well-tested, multi-platform, Python 3 compatible, BSD-licensed library with bleeding edge approaches for multi-label problems, which is actively developed and has a growing user base. We also show that it is faster than

its competition in other languages. It is also entirely compatible with the scientific/machine learning ecosystem of well-known Python libraries which allows it to fill the niche of the missing multi-label classification solution, while also benefits from complementary solutions present in the scikit-learn community, while providing it with efficient implementations of missing classification methods, data stratification, dataset access and manipulation.

We start with a summary of what multi-label classification is in Section 2. Followed by Section 3 where we present related work and a description of the machine learning ecosystem that scikit-multilearn fits in. In Section 4 we describe what scikit-multilearn brings to the community. We evaluate how scikit-multilearn performs in comparison to libraries implemented in other languages in Section 5 and present conclusions and future ideas in Section 6.

2. Multi-label classification

Multi-label classification deals with the problem of assigning a subset of available labels to a given observation. Such problems appear in multiple domains - article and website classification, multimedia annotation, music categorization, the discovery of genomics functionalities and are performed either by transforming a problem into a single/multi-class classification problem or by adapting a single/multi-class method to take multi-label information into account. An excellent introduction to the field has been provided by Tsoumakas et al. (2009).

Madjarov et al. (2012) divide approaches multi-label classification into three groups: method adaptation, problem transformation and ensembles thereof. The first idea is to adapt single-label methods to multi-label classification by modifying label assigning fragments in a single-label method. An example of this is introducing a multi-label version of a decision function in Decision Trees (ex. Kocev et al., 2013).

Problem transformation concentrates on converting the multi-label problem to one or more single-label problems. The most prominent examples include classifying each label separately such as Binary Relevance or Classifier Chains (Read et al., 2009) and treating each label combination as a separate class in one multi-class problem as in the Label Powerset case.

Both of these approaches suffer not only from standard problems of machine learning such as over- and under-fitting but also from the issue of label imbalance (Sun et al., 2009) or numerical anomalies related to label ordering when Bayesian approaches are used for taking correlations into account. Ensemble methods aim to correct this by learning multiple classifiers trained on label subspaces (ex. RAkEL by Tsoumakas et al., 2011a), observation subsets with pruning and replacement, or analyzing various label orderings in chains (ex. Probabilistic Classifier Chains by Dembczynski et al., 2010).

Multi-label embedding techniques emerged as a response the need to cope with a large label space; these include label space dimensionality reduction techniques that turned Most multi-label embedding methods turn multi-label classification into multivariate regression problem followed by a rule-based or classifier-based correction step. Embedding methods also vary by the principle of how the embedding is performed, these include: Principle Label Space Transformation (Tai and Lin, 2012) based on Principal Component Analysis; Conditional Principal Label Space Transformation (Chen and Lin, 2012) with Canonical Com-

ponent Analysis; Feature-aware Implicit Label Space Encoding (Lin et al., 2014) based on matrix decompositions; SLEEC (Bhatia et al., 2015) which uses k-means clustering and per cluster embedding; CLEMS (Huang and Lin, 2017) which performs Multi-Dimensional Scaling (see Kruskal, 1964) with extra multi-label quality measure optimization; and LNEMLC (Szymański et al., 2018) which embeds label networks using network embeddings such as LINE or node2vec. Most of these use linear, ridge or random forest regressors to predict embeddings for unseen samples and a variant of kNN to classify them with labels. CLEMS and LNEMLC are among the best performing multi-label embeddings at this time.

Multi-label advances in recent years also include developments in a sub-area called extreme multi-label where new methods emerged such as deep-learning based domain-specific approaches for: image classification frameworks (Zhao et al., 2015; Wang et al., 2016; Wei et al., 2016) or text (Liu et al., 2017; Yen et al., 2017). The field also includes tree-based (Prabhu and Varma, 2014) or embedding-based (Bhatia et al., 2015) approaches. While extreme multi-label is an interesting task it differs strongly from classical multi-label classification in performance expectations, benchmark datasets and quality measures and falls beyond the main focus of scikit-multilearn but its methods can be used to classify these scale of problems.

3. Related work

As with every applied science, research requires environments for performing experiments. The most prominent multi-label classification stack to date (regarding method count and popularity) is implemented in Java: MULAN (Tsoumakas et al., 2011b) and MEKA (Read et al., 2016). Both depend heavily on the famous WEKA library (Hall et al., 2009) which implements a plethora of classification and regression methods for single-label classification. MULAN and MEKA are large multi-purpose libraries which support not only multi-label classification tasks but also regression and multi-instance/multi-output tasks.

Python's scientific ecosystem's philosophy (Oliphant, 2007) is a different one, instead of having few large multi-purpose libraries, it provides a foundation of libraries that deliver most important approaches and a network of smaller and more specialized libraries that interact thanks to a well-defined API, coding and documentation standards set out for common machine learning tasks such as prediction or data transformation. With numpy (Oliphant, 2006), scipy (Jones et al., 2001) as numerical operation and data structure foundations. The machine-learning foundation provided by scikit-learn (Pedregosa et al., 2011) follows a different approach than WEKA as it does not aim at being a large mono-library where all functionality is gathered. Instead, it concentrates on implementing most cited well-established methods alongside a stable API design.

The rudimentary multi-label methods provided by scikit-learn do not support sparse representations of label matrices apart from the multi output classifier: algorithm adaptation approaches such as k nearest neighbors, CART trees and random forests and perceptrons; problem transformation methods Binary Relevance, Classifier Chains and one-vs-rest or one-vs-all usage of multi-class classifiers such as Support Vector Machines (Hearst et al., 1998). However all of these are the most rudimentary variants of listed methods and have been superseded by more modern approaches. The kNN approach from scikit-learn has been improved by ML-kNN (Zhang and Zhou, 2007), deep learning is used much more

often than perceptrons, and new tree-based classifiers like fastXML (Prabhu and Varma, 2014) outperform traditional random forests of multi-label trees. One-vs-all/rest multi-class classifiers can now be replaced with inherently multi-label variants of the methods, ex. SVM classification scheme has been enhanced by multi-label SVM approaches such as MLTSVM (Chen et al., 2016). Binary Relevance provides worse performance than data-driven label space division (Szymański et al., 2016) or stacking (Montañes et al., 2014). Classifier chains have been extended in multiple ways to overcome artifacts related to sample distribution changes between label orderings either by learning shorter chains via data-driven label space divisions or chain sampling methods (PCC by Dembczynski et al., 2010), (MCC by Read et al., 2013). However scikit-learn's classifiers are extremely useful when used as a base for more comprehensive methods in scikit-mutlilearn such as label space partitioning ensembles or multi-label embedding.

The scikit-learn project also forms a hub of a large network of complementary libraries for more specific tasks, new techniques or emergent sub-fields. Such libraries include among many others scikit-multiflow and imbalanced-learn.

The scikit-multiflow library (Montiel et al., 2018) concerns multi-label streams, which differs from the traditional multi-label setting where a method is trained once to predict data - in multi-label streams the methods are retrained or adopted as new samples arrive. Even though scikit-multiflow provides reimplementations selected methods of scikit-learn, the authors note that scikit-learn's implementations should be applied while using the library. Authors claim that the provided implementations are less optimal (kNN) or masks over the original scikit-learn code for internal purposes. more Modern version of classifier chains (PCC, MCC) and stream classifiers are the core contribution of the library. They are optimized towards partial updating as new evidence arrives. The library is well unit-tested and should be used for multi-label stream tasks which scikit-multilearn does not handle out of the box. However, scikit-multilearn provides more vibrant and various offering of state of the art methods optimized for the classic multi-label problem formulation.

The imbalanced-learn library (Lemaître et al., 2017) is dedicated to overcoming class imbalance problems in single-label tasks by a variety of under/over-sampling strategies. As many problem transformation approaches transform the problem to a multi-class problem imbalanced-learn can be used to improve learning capabilities of scikit-multilearn classifiers by fitting them to resampled data.

Domain and sub-field related deep neural networks with multi-label classification support are also available such as magpie[1], CNN-RNN for images or a variety of different models for biological data[2] for text classification.

Multi-label classification problems can also be addressed in general purse deep learning libraries such as Tensorflow (Abadi et al., 2015) or Keras (Chollet et al., 2015). However, deep neural networks often do not demonstrate an appropriate level of robustness and require significant differences in architecture to accommodate a concrete problem. This introduces an additional level of complication while applying deep learning to multi-label problems. While these libraries are instrumental in their areas, scikit-multilearn is dedicated to providing domain-independent solutions as efficient per domain specialization is rarely possible in one unified code base. However, scikit-multilearn does provide support for using

1. `https://github.com/inspirehep/magpie`
2. `https://qdata.github.io/deep4biomed-web/`

	Java		Python		R	Matlab Octave
	MULAN	MEKA	scikit-learn	scikit-multilearn	util.ml	MLC_toolbox
Algorithm-adaptation methods						
Multi-label SVM				✓		✓
Trees	✓	✓	✓			✓
Nearest Neighbors	✓	✓		✓		✓
Neural	✓	✓	✓	✓		✓
Problem transformation approaches						
To single-class	✓	✓	✓	✓	✓	✓
To multi-class	✓	✓		✓		✓
To deep learning models				✓		
Ensemble approaches						
Stacking	✓	✓	✓	✓	✓	✓
Label space partitioning	✓	✓		✓		✓
Overlapping models	✓	✓		✓		✓
Embeddings				✓		✓
Related tasks						
Multi-label stratification	✓	✓		✓		
Quality measures	✓	✓	✓	P	✓	✓
Multi-label dataset management	✓	✓	P	✓	✓	
Multi-label regression	✓	✓	✓			
Multi-output learning	✓					
Sparse label space representation			P	✓		
Multi-label ARFF support	✓	✓		✓	✓	
Project quality						
End-to-end use cases			✓	✓		
Developer documentation	✓		✓	✓		
API documentation	✓	✓	✓	✓	✓	
Unit-tests	✓	✓	✓	✓	✓	
Multi-platform	✓	✓	✓	✓	✓	P
Continuous integration on all platforms			✓	✓		

Table 1: Comparison of multi-label classification libraries and their features. Letter P denotes partial functionality: scikit-learn can generate multi-label datasets, but does not offer methods to lead them and manipulate them; sparse support is implemented only for Binary Relevance (MultiOutputClassifier). MLC toolbox offers some of its classifiers Windows-only. Output format from scikit-multilearn's classifiers is compatible with scikit-learn's quality measures.

Keras-compatible models for multi-label classification independent of the backend used by Keras (i.e. Tensorflow, CNTK) or PyTorch (via the skorch[3] library).

The scikit-multilearn project fits in the Python community similarly as to how MEKA or MULAN fit to the Java machine learning projects network, however, it is focused on multi-label classification problems to provide state of the art approaches and efficient implementation of more advanced methods.

Recently multi-label classification libraries were made available in R (R Core Team, 2013) and Matlab/Octave (Eaton et al., 2017): utiml (Rivolli and de Carvalho, 2018) R library built on top of the dataset management library mldr (Charte and Charte, 2015) and offers a larger variety of available methods than other R libraries and a more welcoming API; MLC Toolbox (Kimura et al., 2017) provides multi-label methods for Matlab/Octave communities.

Table 1 provides a general point of view on the functionality offered by multi-label classification methods in Java, Python, R and Matlab/Octave.

4. The scikit-multilearn library

The proposed library follows the idea of building a multi-label classification library on top of an existing classification solution. In the case of Python, the obvious choice for a base library is the scipy stack with scikit-learn. The concept of scikit-learn compatible projects (scikits in short, not to mistake with the old scipy notion of scikits), has been present for several years in many formats, most prominently exemplified in scikit-contrib. We follow the ideas of these communities, the scikit-learn API principles and licensing.

The primary goal of scikit-multilearn is to provide an efficient Python implementation of as many multi-label classification algorithms as possible both to the open source community and commercial users of the Python data science stack. With such focus in mind, we concentrate on delivering a library dedicated to provide domain-independent solutions for solving multi-label classification problems, allowing other toolkits to excel in their areas such as: scikit-multiflow when it comes to multi-label streams or imbalanced-learn when it comes to handling imbalance multi-class data or many deep learning approaches described in the previous section.

Extensions of scikit-learn. The scikit-multilearn library extends the multi-label classification offering of scikit-learn by:

- extending the family of classifiers available to the Python community with more advanced algorithm adaptation and problem transformation (label space division) approaches
- implementing multi-label embedding-based classification methods
- providing additional features too specific for scikit-learn such as the general framework for classification based on label space division or the wrapper for MEKA,
- implementing non-classification tools for multi-label classification, including multi-label dataset manipulation from ARFF, loading and saving sparse representations of

3. `https://github.com/dnouri/skorch`

these datasets and the only Python implementations of multi-label data stratification methods (see work by (Sechidis et al., 2011), (Szymański and Kajdanowicz, 2017))

Advanced Multi-label Adapted Algorithms. Within scikit-multilearn we provide implementations of recent multi-label algorithm adaptation methods, that do not meet the selectiveness criteria of scikit-learn due to their novelty. These include:

- ML-ARAM: the hierarchical multi-label adaptive resonance associative map (Benites and Sapozhnikova, 2015)
- BR-kNN and ML-kNN: Binary Relevance kNN (Spyromitros et al., 2008) and multi-label kNN (Zhang and Zhou, 2007) - their scikit-multilearn implementations are built on top of scikit-learn's NearestNeighbors multiclass classifier,
- MLTSVM: the multi-label twin support vector machine (Chen et al., 2016) which does not require a training quadratic number of SVM classifiers.

Problem Transformation with Label Space Division. One of the significant contributions of the library is a framework for performing ensemble classification with different label space division strategies. The library provides ways to divide the label space based on:

- clustering of the label matrix representations, using any of the scikit-learn compatible clusterers (i.e. classes that inherit scikit-learn's `ClusterMixin` base class). The clusterer is executed on a transposed label assignment matrix and detected clustering is used as the label space division for the ensemble method. An example of this approach is the k-means clustering of labels which has been applied by many approaches (Tsoumakas et al., 2008; Yu et al., 2017)).
- communities detected in a network which embeds information about relations between labels. In scikit-multilearn there is a general API for defining graph generators based on the label matrix, such graphs can exploit relations based on co-occurrence, correlation or any other pairwise measure defined over labels. The library supports community detection methods based on three popular graph/network packages in Python: (NetworkX, BSD by Hagberg et al., 2008)), (igraph, GPL by Csardi and Nepusz, 2006), (graphtool, GPL by Peixoto, 2014). An example of this approach is the label space division based on label co-occurrence graphs (Szymański et al., 2016).
- random division, present in the random k-label sets (RAkEL) approaches
- predefined fixed division obtained from expert knowledge

In scikit-multilearn a partitioning and a voting majority ensemble approach are provided, in both a sub-classifier is trained on every label subset from a label space partition. In the partitioning variant, the predicted results are the union of results from each of the classifiers. In the voting ensemble, the label is assigned if the majority of classifiers trained on a label subspace which contain the label have assigned it to a given sample.

Multi-Label Embeddings. Another significant contribution of the library is a framework for multi-label classification using multi-label embeddings and scikit-multilearn is the first Python library to provide a general multi-label embedding scheme with an embedding regressor and embedding-based classifier. Currently two state of the art embedding approaches are provided Cost-Sensitive Label Embedding with Multidimensional Scaling (CLEMS)(Huang and Lin, 2017), Label Network Embeddings for Multi-Label Classification (LNEMLC)(Szymański et al., 2018) alongside a general embedder that can use any scikit-learn manifold learning or dimension reduction to embed the output space.

Compatibility. The scikit-multilearn project aims to be compatible with the Python data science stack. It follows the scikit-learn API and requirements specified in check_estimator code. Multi-label classifiers in scikit-multilearn inherit scikit-learn's `BaseEstimator` and `ClassifierMixin` classes. They can be thus easily incorporated into scikit-learn pipelines and cross-validations, evaluation measures, and feature space transformators. It is easy to use scikit-learn's extensive feature-space manipulation methods, single-label classifiers and classification evaluation functions with scikit-multilearn.

BSD licensing. As innovations in machine learning happen in both academia and companies, it is important to create a library that allows both communities to grow and profit from common work. Thus, scikit-multilearn is released under the BSD license to permit the for-profit ecosystem to use the library while opening the possibility of sharing development and maintenance tasks. While most of scikit-multilearn's functionality is available via BSD licensed libraries, the package also allows using GPL-licensed libraries to be used for label space division such as igraph or graphtool. Importing from `skmultilearn.cluster.igraph` or `skmultilearn.cluster.graphtool` will require your code to submit to GPL licensing requirements.

Access to the Java stack. The library provides scikit-compatible wrapper to reference libraries such as MEKA, MULAN and WEKA through MEKA when a need arises to use methods that have not yet been implemented in Python natively. Using these libraries via scikit-multilearn wrapper does not induce a licensing (GPL-BSD) conflict. Support for MULAN is limited to methods available via MEKA's `meka.classifiers.multilabel.MULAN` classifier[4].

Using Deep Learning models. Scikit-multilearn provides a wrapper that allows using any Keras (Chollet et al., 2015) compatible backend such as Tensorflow (Abadi et al., 2015) CNTK (Seide and Agarwal, 2016) or PyTorch (Paszke et al., 2017), to provide a single-class or multi-class model that can be used to solve multi-label problems through problem transformation approaches.

Sparsity. Multi-label classification problems often do not provide cases for all possible label combinations - in most benchmark sets there are less than 5% labels per row on average. The output space is thus very sparse. As opposed to MULAN and MEKA, scikit-multilearn operates on sparse matrices internally yielding a large memory boost as displayed in Figure 1.

4. `http://meka.sourceforge.net/api-1.9/meka/classifiers/multilabel/MULAN.html#setMethod-java.lang.String-`

Using scikit-multilearn. The steps for using scikit-multilearn are very simple: loading the data, selecting the method and performing classification. The library supports loading matrices from all well-established formats thanks to scipy and operates on scipy/numpy representations internally, converting to dense matrices or lists of lists only if required by a scikit-base classifier, if one is employed in the problem transformation scenario.

Other output types and subproblems. Scikit-multilearn does not support multi-label regression or multi-output prediction at the moment. Multi-label regression is considered for future releases, while multi-output approaches will not be supported due to scikit-multilearn's concentration on efficient sparse matrix usage throughout the codebase. Extreme Multi-label Classification is not the main focus of scikit-multilearn but both the label network embedding methods and label space partitioning methods are capable of handling extreme multi-label datasets. Keras-based deep learning models can also be deployed with the label partitioning strategy to solve extreme multi-label problems.

Structure of scikit-multilearn. Scikit-multilearn provides three subpackages following the established categorization of multi-label methods: method adaptation approach (in `skmultilearn.adapt`), problem transformation approach (in `skmultilearn.pt`) and ensembles of the two (in `skmultilearn.ensemble`). The label space division methods and label relationship graph builders are present in `skmultilearn.cluster`. To use multi-label-adapted version of a single-label method one only needs to import it and instantiate an object of the class. A problem transformation approach takes a scikit-learn compatible base classifier and optional information whether the base classifier supports sparse input. Ensemble methods require a classifier and relevant method's parameters. The classifier is trained using the fit method, which takes the training input and output matrices, and classification is performed using the predict method on test observations - just as in scikit-learn. The MEKA wrapper provides a scikit-learn compatible classifier class in `skmultilearn.meka`. Dataset download, loading and manipulation tools are available in `skmultilearn.dataset`. Multi-label data stratification methods and dataset fold division quality measures are available in `skmultilearn.model_selection`. Multi-label embedding approaches are located in `skmultilearn.embeddings`.

4.1. Documentation and Availability

Scikit-multilearn is hosted on GitHub and managed via the fork and pull-request paradigm. Both user and developer documentation is available at `http://scikit.ml/`. After 4 years of development, in July 2018, the library has reached a stable 0.1.0 release. The newest version - 0.2.0 - was released in December 2018. The code is released under BSD licence using GitHub and releases are available via PyPI. Implementation follows PEP8 and is maintained with a high level of test coverage (82% for the 0.2.0 release). The development undergoes continuous integration on Windows, Ubuntu Linux and Mac OS X, with Python 2.7 and Python 3.3. At the moment of the 0.2.0 release scikit-multilearn has been starred 242 times and forked 72 times. Development is managed on the GitHub repository `scikit-multilearn`[5] and is accompanied with communication and discussion in the Slack

5. `https://github.com/scikit-multilearn/scikit-multilearn`

channel[6]. All commits undergo continuous testing on Windows, Ubuntu and Mac OS X, both under Python 2.7 and 3.3. The library also has its tag[7] on StackOverflow.

5. Benchmark

We found that scikit-multilearn is faster than its competition. We have tested MEKA, MULAN and scikit-multilearn on 12 well-cited benchmark multi-label classification datasets using two comparison scenarios: Binary Relevance and Label Powerset. The two selected problem transformation approaches are widely used both in regular classification tasks and as the base for more sophisticated methods. We did not test algorithm adaptation methods as there are no algorithm adaptation methods present in all three libraries. We use these methods to illustrate two aspects of the classification performance of the libraries: the cost of using many classifiers with splitting operations performed on the label space matrix and the cost of using a single classifier which requires to access all label combinations to perform the transformation. To minimize the impact of base classifiers, we have decided to use a fast Random Forest base classifier with 10 trees. As Octave does not provide Matlab's random forest implementation, we used the one provided by the shogun toolbox (Sonnenburg et al., 2017). We have checked the classification quality and did not find significant differences between Hamming Loss, Jaccard and Accuracy scores between the outputs. The benchmark is performed with scikit-multilearn 0.1.0, MEKA 1.9.2, MULAN 1.5.0, scikit-learn 0.19.2, Octave 4.2.2, shogun 6.1.3, MLC Toolbox for Matlab/Octave code from GitHub master commit hash `e798779`, R 3.4.1, utiml 0.1.2. As utiml does not provide a Label Powerset implementation, we do not evaluate it in this subexperiment.

Figure 1 presents the time and memory required to perform classification, including loading the ARFF dataset (or Matlab export in case of the toolbox) and measuring errors, i.e. a complete classification use case scenario. As different datasets require a different amount of time and memory we decided to normalize the charts. The results on the chart are normalized for each dataset separately. To normalize we calculated the median of every library's time or memory performance and selected the highest of the medians in the dataset as the normalization point, i.e. 100% is the worst median performance. We present the best, median, and worst performance of each library per dataset, normalized performance with the worst median performance on that dataset. The closer the library is to point 0, the better it performed, thus the smaller area inside the library curve, the better.

All the libraries were forced to use a single core using the `taskset` command to minimize parallelization effects on the comparison. Time and memory results were obtained using the `time -v` command and represent User time, and Maximum resident set size respectively. All results taken into consideration reported that 100% of their CPU core had been assigned to the process which performed the classification scenario.

We notice that in both classification schemes scikit-multilearn always uses less or, in a few edge cases the same amount of, memory than MEKA or MULAN due to its sparse matrix support. In most cases scikit-multilearn also operates faster than MEKA, MULAN.

When it comes to label space division approaches modeled in the experiment by Binary Relevance, scikit-multilearn is the most efficient choice on every data set. With Label Pow-

6. `https://scikit-ml.slack.com`
7. `https://stackoverflow.com/tags/scikit-multilearn/info`

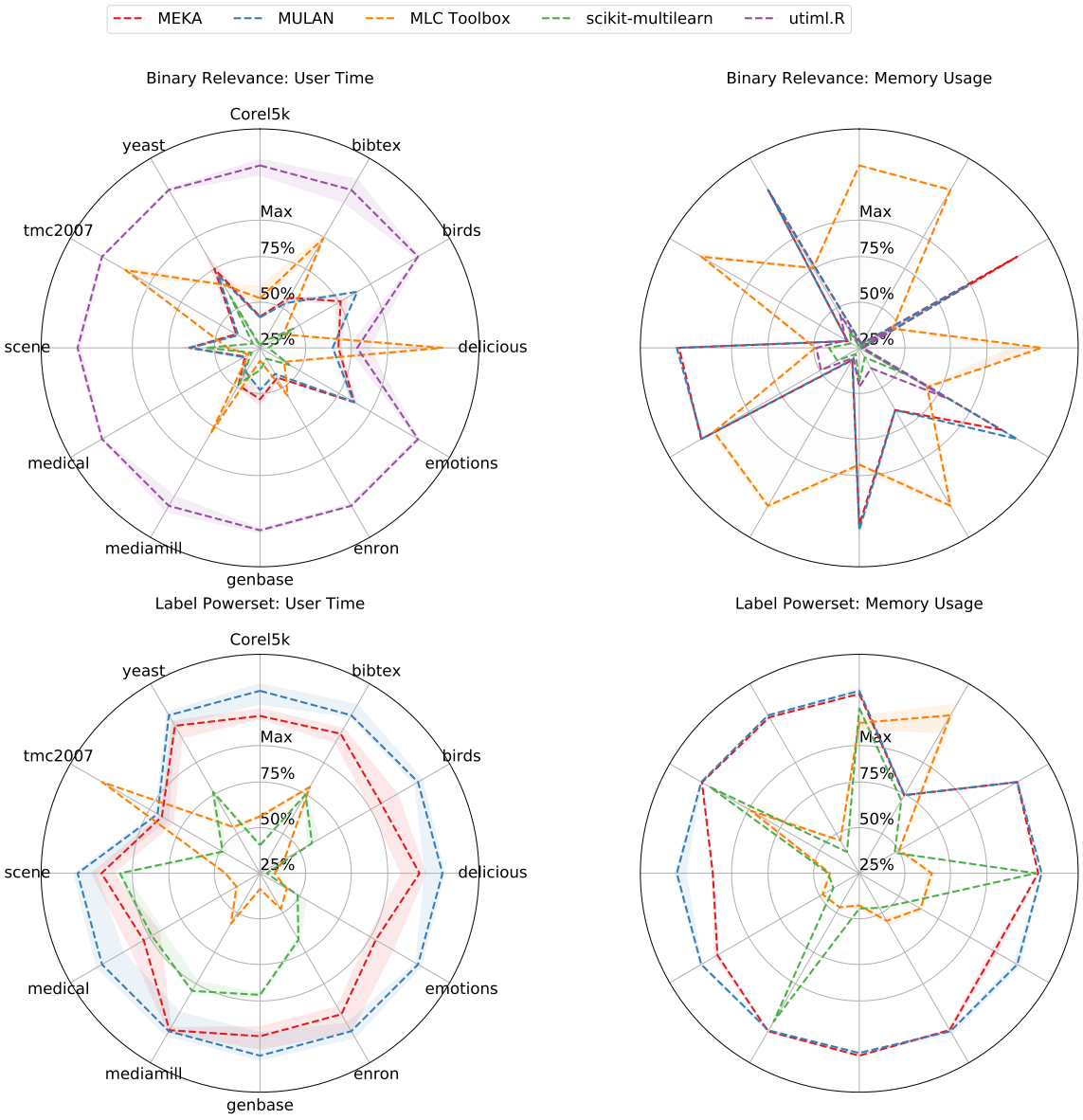

Figure 1: User running time (s) and memory usage ranks of utiml, MLC Toolbox, scikit-multilearn, MEKA and MULAN, with RandomForest, and two different multi-label classifiers.

erset multi-class transformation it is in most cases more efficient than MEKA and MULAN and outperforms the MLC Toolbox on datasets with larger numbers of label combinations while on smaller data sets the MLC Toolbox tends to be more efficient.

We have also compared Binary Relevance from scikit-learn to Binary Relevance from scikit-multilearn. The only difference between the two implementations is the fact that

scikit-learn does not convert the output matrix to a Compressed Sparse Column matrix[8] while scikit-multilearn does. The observed results were that scikit-learn's implementation was always slightly faster than scikit-multilearn's, while it always used a slightly larger amount of memory, both changes can be attributed to a difference between formats used for the output space. We, therefore, do not report this comparison in extensive detail.

As there is no statistical procedure that allows performing a non-parametric repeated measure tests on multiple measurements per test, we decided to compare the worst case time/memory usage of scikit-multilearn against the best case MEKA/MULAN/utiml/MLC

8. `https://docs.scipy.org/doc/scipy/reference/generated/scipy.sparse.csc_matrix.html`

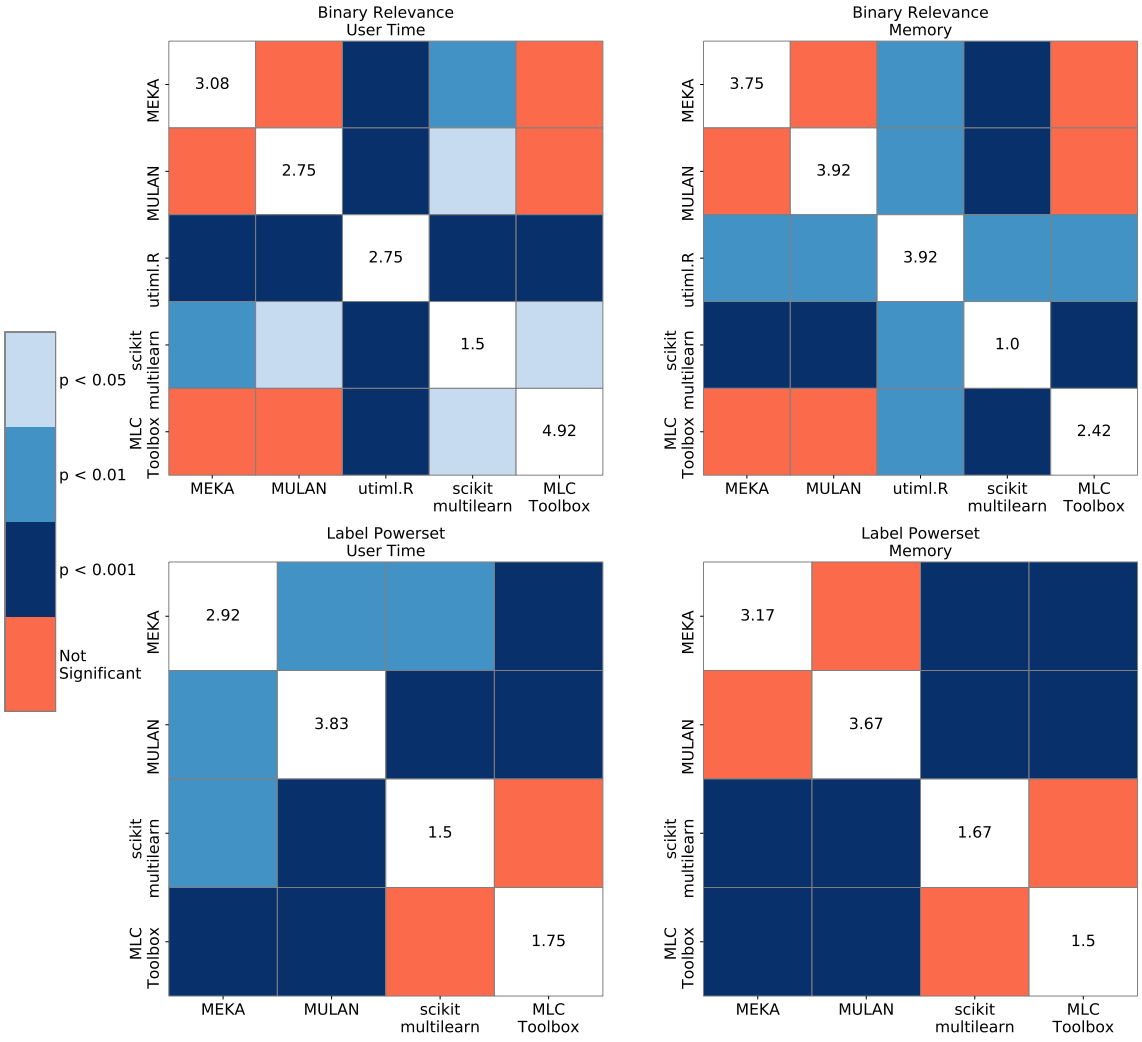

Figure 2: User running time (s) and memory usage ranks of utiml, MLC Toolbox, scikit-multilearn, MEKA and MULAN, with RandomForest, with RandomForest, and different multi-label classifiers. There are mean ranks on diagonal.

Toolbox performance. We used the Quade test with Bergmann-Hommel post hoc multiple hypothesis testing scenario (as laid out by García et al., 2010), the significances of rank difference and mean ranks of each libraries performance are provided in Figure 2.

We see that even in this negative averaging scenario scikit-multilearn is statistically significantly more efficient the other libraries when label space division approach (Binary Relevance) is deployed.

When Label Powerset multi-class transformation is used both scikit-multilearn and MLC Toolbox perform better than MULAN/MEKA with statistical significance. In terms of speed scikit-multilearn ranks slightly higher than the MLC Toolbox but consumes a little more memory. The differences between scikit-multilearn and the MLC Toolbox in this approach are not statistically significant.

6. Conclusions

The presented library - scikit-multilearn - is the most extensive scikit-learn compatible multi-label classification library. It provides implementations of both the most popular algorithms and new families of methods such as network-based label space division approaches. It is fast thanks to performing transformations on sparse matrices internally and using well-optimized methods from scikit-learn as base classifiers, and also integrates well with the Python data science stack of scipy. It also provides wrappers to Keras models which allows quick adaptation of deep learning approaches to multi-label classification via problem-transformation approaches, and MEKA/WEKA with parts of MULAN - the standard Java classification stack - and allows easy use of those methods with the rest of the Python stack.

Acknowledgments

We would like to thank Wojciech Stachowski for initial versions of BR/ML-kNN & Classifier Chains implementations, Christian Schulze for asking multiple questions, reporting and fixing bugs, Felipe Almeida for bug fixes and testing, Fernando Benites for providing the implementation of ML-ARAM (Brucker et al., 2011), and Jose Perez-Parras Toledano for his implementation of balanced k-means clustering and other contributions. The work was partially supported by The National Science Centre the research projects no. 2016/21/N/ST6/02382 and 2016/21/D/ST6/02948, by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 691152 (RENOIR); the Polish Ministry of Science and Higher Education fund for supporting internationally co-financed projects in 2016-2019 (agreement no. 3628/H2020/2016/2) and by the Faculty of Computer Science and Management, Wrocław University of Science and Technology statutory funds.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefow-

icz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL `http://tensorflow.org/`. Software available from tensorflow.org.

F. Benites and E. Sapozhnikova. Haram: A hierarchical aram neural network for large-scale text classification. In *2015 IEEE International Conference on Data Mining Workshop (ICDMW)*, pages 847–854, Nov 2015. doi: 10.1109/ICDMW.2015.14.

Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. Sparse local embeddings for extreme multi-label classification. In *Advances in Neural Information Processing Systems*, pages 730–738, 2015.

Florian Brucker, Fernando Benites, and Elena Sapozhnikova. Multi-label classification and extracting predicted class hierarchies. *Pattern Recognition*, 44(3):724 – 738, 2011.

Francisco Charte and David Charte. Working with multilabel datasets in R: The mldr package. *The R Journal*, 7(2):149–162, December 2015. URL `https://journal.r-project.org/archive/2015-2/charte-charte.pdf`.

Wei-Jie Chen, Yuan-Hai Shao, Chun-Na Li, and Nai-Yang Deng. Mltsvm: a novel twin support vector machine to multi-label learning. *Pattern Recognition*, 52:61–74, 2016.

Yao-Nan Chen and Hsuan-Tien Lin. Feature-aware label space dimension reduction for multi-label classification. In *Advances in Neural Information Processing Systems*, pages 1529–1537, 2012.

François Chollet et al. Keras. `https://keras.io`, 2015.

Gabor Csardi and Tamas Nepusz. The igraph software package for complex network research. *InterJournal*, Complex Systems:1695, 2006. URL `http://igraph.org`.

Krzysztof Dembczynski, Weiwei Cheng, and Eyke Hüllermeier. Bayes optimal multilabel classification via probabilistic classifier chains. In *ICML*, volume 10, pages 279–286, 2010.

John W. Eaton, David Bateman, Søren Hauberg, and Rik Wehbring. *GNU Octave version 4.2.1 manual: a high-level interactive language for numerical computations*, 2017.

Salvador García, Alberto Fernández, Julián Luengo, and Francisco Herrera. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. *Information Sciences*, 180 (10):2044–2064, 2010.

Aric Hagberg, Pieter Swart, and Daniel S Chult. Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States), 2008.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. The WEKA data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November 2009. ISSN 1931-0145.

Marti A. Hearst, Susan T Dumais, Edgar Osuna, John Platt, and Bernhard Scholkopf. Support vector machines. *IEEE Intelligent Systems and their applications*, 13(4):18–28, 1998.

Kuan-Hao Huang and Hsuan-Tien Lin. Cost-sensitive label embedding for multi-label classification. *Machine Learning*, 106(9-10):1725–1746, 2017.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL `http://www.scipy.org/`. Online; Accessed 2018-11-30.

Keigo Kimura, Lu Sun, and Mineichi Kudo. Mlc toolbox: A matlab/octave library for multi-label classification. *arXiv preprint arXiv:1704.02592*, 2017.

Dragi Kocev, Celine Vens, Jan Struyf, and Sašo Džeroski. Tree ensembles for predicting structured outputs. *Pattern Recognition*, 46(3):817–833, 2013.

Joseph B Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

Guillaume Lemaître, Fernando Nogueira, and Christos K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research*, 18(17):1–5, 2017. URL `http://jmlr.org/papers/v18/16-365.html`.

Zijia Lin, Guiguang Ding, Mingqing Hu, and Jianmin Wang. Multi-label classification via feature-aware implicit label space encoding. In *International conference on machine learning*, pages 325–333, 2014.

Jingzhou Liu, Wei-Cheng Chang, Yuexin Wu, and Yiming Yang. Deep learning for extreme multi-label text classification. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 115–124. ACM, 2017.

Gjorgji Madjarov, Dragi Kocev, Dejan Gjorgjevikj, and Sašo Džeroski. An extensive experimental comparison of methods for multi-label learning. *Pattern Recognition*, 45(9): 3084–3104, September 2012.

Elena Montañes, Robin Senge, Jose Barranquero, José Ramón Quevedo, Juan José del Coz, and Eyke Hüllermeier. Dependent binary relevance models for multi-label classification. *Pattern Recognition*, 47(3):1494–1508, 2014.

Jacob Montiel, Jesse Read, Albert Bifet, and Talel Abdessalem. Scikit-Multiflow: A Multi-output Streaming Framework. *CoRR*, abs/1807.04662, 2018.

Travis E Oliphant. *A guide to NumPy*, volume 1. Trelgol Publishing USA, 2006.

Travis E Oliphant. Python for scientific computing. *Computing in Science & Engineering*, 9(3), 2007.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. In *NIPS-W*, 2017.

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. *Journal of machine learning research*, 12(Oct):2825–2830, 2011.

Tiago P. Peixoto. The graph-tool python library. *figshare*, 2014. doi: 10.6084/m9.figshare.1164194. URL `http://figshare.com/articles/graph_tool/1164194`.

Yashoteja Prabhu and Manik Varma. Fastxml: A fast, accurate and stable tree-classifier for extreme multi-label learning. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 263–272. ACM, 2014.

R Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2013.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 254–269. Springer, 2009.

Jesse Read, Luca Martino, and David Luengo. Efficient monte carlo optimization for multi-label classifier chains. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3457–3461. IEEE, 2013.

Jesse Read, Peter Reutemann, Bernhard Pfahringer, and Geoff Holmes. MEKA: A multi-label/multi-target extension to Weka. *Journal of Machine Learning Research*, 17(21):1–5, 2016.

Adriano Rivolli and Andre C. P. L. F. de Carvalho. The utiml Package: Multi-label Classification in R. *The R Journal*, 2018. URL `https://journal.r-project.org/archive/2018/RJ-2018-041/index.html`.

Konstantinos Sechidis, Grigorios Tsoumakas, and Ioannis Vlahavas. On the stratification of multi-label data. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 145–158. Springer, 2011.

Frank Seide and Amit Agarwal. Cntk: Microsoft's open-source deep-learning toolkit. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 2135–2135, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4232-2. doi: 10.1145/2939672.2945397. URL `http://doi.acm.org/10.1145/2939672.2945397`.

Soeren Sonnenburg, Heiko Strathmann, Sergey Lisitsyn, Viktor Gal, Fernando J. Iglesias García, Wu Lin, Soumyajit De, Chiyuan Zhang, frx, tklein23, Evgeniy Andreev, Jonas-Behr, sploving, Parijat Mazumdar, Christian Widmer, Pan Deng / Zora, Giovanni De Toni, Saurabh Mahindre, Abhijeet Kislay, Kevin Hughes, Roman Votyakov, khalednasr, Sanuj Sharma, Alesis Novik, Abinash Panda, Evangelos Anagnostopoulos, Liang Pang, Alex Binder, serialhex, and Björn Esser. shogun-toolbox/shogun: Shogun 6.1.0, November 2017.

Eleftherios Spyromitros, Grigorios Tsoumakas, and Ioannis Vlahavas. An empirical study of lazy multilabel classification algorithms. In *Hellenic conference on artificial intelligence*, pages 401–406. Springer, 2008.

Yanmin Sun, Andrew KC Wong, and Mohamed S Kamel. Classification of imbalanced data: A review. *International Journal of Pattern Recognition and Artificial Intelligence*, 23(04): 687–719, 2009.

Piotr Szymański and Tomasz Kajdanowicz. A Network Perspective on Stratification of Multi-Label Data. In *First International Workshop on Learning with Imbalanced Domains: Theory and Applications, LIDTA 2017, 22 September 2017, ECML-PKDD, Skopje, Macedonia, Proceedings of Machine Learning Research*, volume 74, pages 22–35, 2017.

Piotr Szymański, Tomasz Kajdanowicz, and Kristian Kersting. How is a data-driven approach better than random choice in label space division for multi-label classification? *Entropy*, 18(8):282, 2016.

Piotr Szymański, Tomasz Kajdanowicz, and Nitesh Chawla. Lnemlc: Label network embeddings for multi-label classifiation, 2018.

Farbound Tai and Hsuan-Tien Lin. Multilabel classification with principal label space transformation. *Neural Computation*, 24(9):2508–2542, 2012.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Effective and efficient multilabel classification in domains with large number of labels. In *Proc. ECML/PKDD 2008 Workshop on Mining Multidimensional Data (MMD'08)*, volume 21, pages 53–59. sn, 2008.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer, 2009.

Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 23(7): 1079–1089, 2011a.

Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, Jozef Vilcek, and Ioannis Vlahavas. Mulan: A java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414, 2011b.

Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294, 2016.

Yunchao Wei, Wei Xia, Min Lin, Junshi Huang, Bingbing Ni, Jian Dong, Yao Zhao, and Shuicheng Yan. Hcp: A flexible cnn framework for multi-label image classification. *IEEE transactions on pattern analysis and machine intelligence*, 38(9):1901–1907, 2016.

Ian EH Yen, Xiangru Huang, Wei Dai, Pradeep Ravikumar, Inderjit Dhillon, and Eric Xing. Ppdsparse: A parallel primal-dual sparse method for extreme classification. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 545–553. ACM, 2017.

Zhilou Yu, Hong Hao, Weipin Zhang, and Hongjun Dai. A classifier chain algorithm with k-means for multi-label classification on clouds. *Journal of Signal Processing Systems*, 86 (2-3):337–346, 2017.

Min-Ling Zhang and Zhi-Hua Zhou. Ml-knn: A lazy learning approach to multi-label learning. *Pattern recognition*, 40(7):2038–2048, 2007.

Fang Zhao, Yongzhen Huang, Liang Wang, and Tieniu Tan. Deep semantic ranking based hashing for multi-label image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1556–1564, 2015.

set	library	count	mean	std	min	25%	50%	75%	max
Corel5k	MEKA	10	329.35	6.93	311.28	328.73	330.65	331.59	336.41
	MLC-Toolbox	10	540.39	64.40	469.11	489.02	524.54	584.42	652.68
	MULAN	10	323.59	5.40	316.28	321.09	322.94	326.08	332.94
	scikit-multilearn	10	24.05	0.93	22.60	23.18	24.71	24.77	24.85
	utiml.R	10	1928.82	54.03	1825.97	1906.33	1929.56	1961.09	1995.39
bibtex	MEKA	10	317.91	7.41	307.67	311.13	319.67	325.21	326.59
	MLC-Toolbox	10	696.19	34.06	633.80	680.56	696.30	718.40	749.24
	MULAN	10	288.15	6.33	284.30	284.87	285.51	286.77	303.88
	scikit-multilearn	10	41.05	1.74	37.19	41.17	41.81	42.00	42.38
	utiml.R	10	1001.93	43.17	925.99	1005.15	1007.94	1010.66	1075.69
birds	MEKA	10	3.63	0.16	3.38	3.56	3.60	3.68	3.90
	MLC-Toolbox	10	1.05	0.03	1.02	1.02	1.04	1.08	1.10
	MULAN	10	4.30	0.10	4.09	4.27	4.34	4.38	4.40
	scikit-multilearn	10	1.50	0.05	1.45	1.46	1.50	1.54	1.58
	utiml.R	10	7.04	0.04	6.97	7.02	7.04	7.06	7.10
delicious	MEKA	10	5291.10	76.51	5193.85	5257.89	5288.31	5298.34	5483.73
	MLC-Toolbox	10	12294.52	519.71	11082.67	12141.24	12302.92	12510.35	13018.59
	MULAN	10	4902.66	37.00	4861.75	4879.77	4896.75	4901.54	4969.57
	scikit-multilearn	10	549.79	18.06	525.48	529.89	562.26	563.51	565.50
	utiml.R	10	6680.74	539.44	5930.92	6453.18	6549.90	7168.62	7381.27
emotions	MEKA	10	2.31	0.04	2.20	2.31	2.32	2.33	2.35
	MLC-Toolbox	10	0.60	0.02	0.58	0.59	0.60	0.62	0.63
	MULAN	10	2.32	0.06	2.25	2.28	2.31	2.34	2.43
	scikit-multilearn	10	0.75	0.06	0.67	0.72	0.72	0.74	0.86
	utiml.R	10	3.89	0.03	3.84	3.89	3.89	3.91	3.93
enron	MEKA	10	17.74	0.27	17.37	17.58	17.72	17.86	18.21
	MLC-Toolbox	10	28.62	0.27	28.31	28.45	28.53	28.76	29.19
	MULAN	10	15.53	0.20	15.32	15.35	15.48	15.69	15.84
	scikit-multilearn	10	6.40	0.23	5.82	6.44	6.48	6.52	6.57
	utiml.R	10	93.75	0.55	92.72	93.34	93.99	94.13	94.27
genbase	MEKA	10	11.94	0.60	11.19	11.71	11.80	12.18	13.32
	MLC-Toolbox	10	2.95	0.07	2.86	2.92	2.95	3.00	3.09
	MULAN	10	9.69	0.21	9.39	9.56	9.68	9.74	10.15
	scikit-multilearn	10	5.01	0.22	4.52	4.89	5.12	5.16	5.18
	utiml.R	10	41.89	0.33	41.48	41.64	41.88	41.97	42.49
mediamill	MEKA	10	607.94	79.02	448.97	624.48	638.55	655.44	662.81
	MLC-Toolbox	10	1446.38	73.95	1285.85	1419.56	1463.09	1504.47	1518.34
	MULAN	10	424.29	23.80	415.32	416.04	416.89	417.42	491.95
	scikit-multilearn	10	595.14	19.48	566.27	576.67	608.73	610.54	611.08
	utiml.R	10	2701.30	102.56	2536.77	2606.49	2704.90	2800.31	2807.30
medical	MEKA	10	6.66	0.15	6.43	6.54	6.65	6.78	6.87
	MLC-Toolbox	10	5.06	0.14	4.83	4.99	5.08	5.14	5.24
	MULAN	10	7.17	0.32	6.90	7.01	7.08	7.14	8.04
	scikit-multilearn	10	3.65	0.07	3.58	3.58	3.64	3.71	3.76
	utiml.R	10	69.57	0.19	69.35	69.44	69.59	69.63	69.99
scene	MEKA	10	4.86	0.16	4.72	4.73	4.83	4.89	5.23
	MLC-Toolbox	10	2.21	0.03	2.16	2.19	2.21	2.23	2.26
	MULAN	10	4.82	0.27	4.50	4.59	4.76	5.06	5.26
	scikit-multilearn	10	3.68	0.09	3.56	3.60	3.70	3.76	3.78
	utiml.R	10	12.47	0.10	12.27	12.43	12.48	12.55	12.59
tmc2007	MEKA	10	154.56	3.17	147.05	153.69	155.76	156.33	158.31
	MLC-Toolbox	10	874.41	37.77	810.27	859.82	884.38	897.00	930.94
	MULAN	10	146.53	5.67	143.46	143.76	143.88	144.15	157.80
	scikit-multilearn	10	47.94	0.98	46.83	47.02	47.95	48.85	48.99
	utiml.R	10	1039.77	2.68	1036.34	1038.47	1038.92	1041.36	1044.98
yeast	MEKA	10	6.73	0.50	6.21	6.47	6.64	6.85	8.00
	MLC-Toolbox	10	5.29	0.05	5.20	5.26	5.30	5.33	5.35
	MULAN	10	6.10	0.14	5.91	6.05	6.09	6.14	6.43
	scikit-multilearn	10	4.35	0.11	4.23	4.25	4.34	4.45	4.48
	utiml.R	10	13.24	0.06	13.15	13.19	13.24	13.28	13.33

Table 2: User time of library running Binary Relevance for each of the datasets.

set	library	count	mean	std	min	25%	50%	75%	max
Corel5k	MEKA	10	841.26	5.66	831.88	837.45	842.15	845.64	849.58
	MLC-Toolbox	10	20062.98	24.13	20032.54	20042.36	20058.43	20085.25	20100.32
	MULAN	10	829.86	4.53	824.72	826.75	827.35	833.10	837.73
	scikit-multilearn	10	168.79	0.05	168.72	168.75	168.78	168.80	168.88
	utiml.R	10	506.28	1.88	502.55	505.85	506.18	507.63	508.95
bibtex	MEKA	10	790.58	5.33	778.96	791.70	793.06	793.50	794.51
	MLC-Toolbox	10	30794.48	79.82	30670.80	30734.07	30794.87	30858.47	30920.78
	MULAN	10	787.87	3.13	785.26	786.10	786.86	788.13	795.67
	scikit-multilearn	10	445.47	0.07	445.40	445.42	445.44	445.53	445.59
	utiml.R	10	992.02	3.12	983.88	992.73	992.82	992.86	994.79
birds	MEKA	10	535.83	1.69	534.09	534.66	535.35	537.34	538.52
	MLC-Toolbox	10	112.99	0.72	111.29	113.08	113.19	113.34	113.63
	MULAN	10	366.19	2.80	361.58	364.39	366.54	368.41	369.66
	scikit-multilearn	10	57.03	0.01	57.01	57.02	57.03	57.04	57.05
	utiml.R	10	108.24	0.00	108.24	108.24	108.24	108.24	108.25
delicious	MEKA	10	2943.83	7.09	2934.48	2937.81	2943.49	2949.36	2954.81
	MLC-Toolbox	10	140445.37	8393.01	119607.80	139070.95	144463.25	145189.90	145481.35
	MULAN	10	2902.95	4.46	2896.79	2899.11	2903.03	2906.44	2909.23
	scikit-multilearn	10	1301.89	1.44	1299.53	1301.04	1301.73	1302.37	1304.74
	utiml.R	10	2401.63	12.06	2380.95	2394.94	2401.16	2408.31	2421.12
emotions	MEKA	10	163.80	2.01	160.37	162.80	164.58	165.30	165.70
	MLC-Toolbox	10	78.75	0.63	78.07	78.41	78.69	78.76	80.36
	MULAN	10	183.30	1.63	180.41	183.02	183.24	184.68	185.20
	scikit-multilearn	10	50.80	0.01	50.78	50.79	50.80	50.80	50.82
	utiml.R	10	100.75	0.88	100.30	100.30	100.30	100.30	102.32
enron	MEKA	10	664.52	3.56	661.21	662.56	663.84	665.20	673.58
	MLC-Toolbox	10	1685.68	18.74	1657.18	1671.94	1687.40	1697.67	1716.76
	MULAN	10	665.95	3.25	662.15	663.86	664.72	667.29	673.14
	scikit-multilearn	10	102.01	0.01	102.00	102.00	102.01	102.02	102.02
	utiml.R	10	219.62	4.54	216.90	216.98	217.09	220.97	229.98
genbase	MEKA	10	662.21	1.73	658.96	661.69	662.61	663.14	664.42
	MLC-Toolbox	10	438.85	6.97	430.62	433.28	436.34	443.71	450.86
	MULAN	10	682.95	3.31	677.60	681.85	682.10	683.88	690.77
	scikit-multilearn	10	125.75	0.02	125.73	125.74	125.75	125.76	125.79
	utiml.R	10	146.65	0.07	146.58	146.58	146.64	146.72	146.72
mediamill	MEKA	10	1122.80	4.01	1119.86	1121.22	1121.39	1122.73	1133.81
	MLC-Toolbox	10	14529.87	85.39	14416.90	14479.99	14506.57	14551.84	14717.54
	MULAN	10	1057.10	29.06	974.44	1065.56	1065.99	1066.99	1067.45
	scikit-multilearn	10	572.13	0.31	571.48	572.05	572.11	572.34	572.60
	utiml.R	10	909.74	3.17	904.12	911.34	911.34	911.34	911.34
medical	MEKA	10	645.37	1.23	642.86	645.07	645.28	646.36	646.94
	MLC-Toolbox	10	593.72	4.43	588.30	590.20	593.54	596.65	602.43
	MULAN	10	642.04	2.46	639.50	639.79	642.54	642.62	647.13
	scikit-multilearn	10	89.29	0.02	89.26	89.28	89.29	89.30	89.31
	utiml.R	10	158.05	2.52	156.46	156.46	156.51	158.93	162.40
scene	MEKA	10	628.31	1.92	624.99	627.08	628.97	629.00	631.96
	MLC-Toolbox	10	154.23	2.95	150.09	152.42	153.38	155.38	159.36
	MULAN	10	637.62	1.87	635.83	636.15	636.59	639.01	640.55
	scikit-multilearn	10	105.90	0.04	105.88	105.88	105.89	105.90	106.02
	utiml.R	10	149.66	0.67	149.44	149.44	149.44	149.44	151.46
tmc2007	MEKA	10	804.08	2.02	801.53	802.19	804.07	805.65	806.88
	MLC-Toolbox	10	11013.90	142.29	10827.76	10946.62	10962.75	11055.96	11321.70
	MULAN	10	823.86	1.89	821.51	822.77	823.19	824.86	827.12
	scikit-multilearn	10	536.68	1.53	534.21	535.62	537.36	537.78	538.13
	utiml.R	10	918.84	13.66	912.77	913.02	914.89	915.48	955.07
yeast	MEKA	10	627.35	3.28	623.79	626.16	626.79	626.86	636.20
	MLC-Toolbox	10	319.40	1.94	316.86	318.17	319.49	320.00	323.11
	MULAN	10	631.91	2.94	627.02	630.77	632.25	633.77	635.93
	scikit-multilearn	10	69.11	0.01	69.09	69.10	69.11	69.11	69.12
	utiml.R	10	132.98	0.63	132.78	132.78	132.78	132.78	134.76

Table 3: Maximum memory usage (in kilobytes) of library running Binary Relevance for each of the datasets.

set	library	count	mean	std	min	25%	50%	75%	max
Corel5k	MEKA	10	47.93	1.06	46.67	47.08	47.60	48.82	49.81
	MLC-Toolbox	10	17.19	0.24	16.86	16.98	17.21	17.34	17.54
	MULAN	10	54.92	1.88	51.15	54.24	55.19	56.07	57.34
	scikit-multilearn	10	8.56	0.33	8.15	8.28	8.48	8.81	9.06
bibtex	MEKA	10	46.60	1.37	43.60	45.98	46.64	47.56	48.48
	MLC-Toolbox	10	28.83	0.47	28.15	28.62	28.84	29.08	29.70
	MULAN	10	53.81	1.66	52.24	52.62	52.81	55.21	56.44
	scikit-multilearn	10	26.36	0.14	26.23	26.25	26.32	26.40	26.66
birds	MEKA	10	2.03	0.17	1.82	1.88	1.98	2.21	2.25
	MLC-Toolbox	10	0.40	0.01	0.38	0.40	0.40	0.40	0.41
	MULAN	10	2.60	0.04	2.53	2.57	2.58	2.63	2.66
	scikit-multilearn	10	0.85	0.05	0.77	0.84	0.85	0.86	0.97
delicious	MEKA	10	1406.20	86.09	1264.17	1357.53	1427.58	1480.17	1486.04
	MLC-Toolbox	10	131.27	4.79	123.27	128.79	130.55	135.06	137.92
	MULAN	10	1606.88	63.40	1478.83	1623.74	1632.84	1642.51	1651.81
	scikit-multilearn	10	59.40	1.15	57.64	58.30	59.73	60.41	60.68
emotions	MEKA	10	1.63	0.11	1.49	1.55	1.58	1.72	1.82
	MLC-Toolbox	10	0.37	0.01	0.36	0.36	0.36	0.37	0.38
	MULAN	10	2.16	0.03	2.13	2.14	2.16	2.18	2.21
	scikit-multilearn	10	0.53	0.06	0.47	0.48	0.52	0.59	0.63
enron	MEKA	10	7.54	0.43	7.07	7.21	7.46	7.69	8.39
	MLC-Toolbox	10	1.91	0.03	1.87	1.89	1.90	1.91	1.98
	MULAN	10	8.39	0.26	7.94	8.22	8.36	8.64	8.71
	scikit-multilearn	10	3.54	0.09	3.42	3.48	3.52	3.63	3.67
genbase	MEKA	10	3.94	0.19	3.67	3.81	3.90	4.10	4.21
	MLC-Toolbox	10	0.38	0.02	0.36	0.36	0.38	0.39	0.42
	MULAN	10	4.32	0.19	3.81	4.32	4.36	4.42	4.47
	scikit-multilearn	10	2.92	0.04	2.86	2.90	2.91	2.95	2.98
mediamill	MEKA	10	646.99	7.33	637.66	640.00	648.32	652.74	657.56
	MLC-Toolbox	10	210.37	4.56	205.49	207.03	207.94	214.65	218.22
	MULAN	10	647.64	26.74	573.99	650.70	652.64	658.56	670.37
	scikit-multilearn	10	479.39	15.57	445.40	475.34	486.05	490.02	491.12
medical	MEKA	10	2.74	0.19	2.36	2.67	2.74	2.78	3.05
	MLC-Toolbox	10	0.55	0.02	0.54	0.54	0.55	0.56	0.59
	MULAN	10	3.68	0.18	3.21	3.70	3.72	3.76	3.84
	scikit-multilearn	10	2.52	0.08	2.30	2.52	2.54	2.55	2.59
scene	MEKA	10	3.36	0.14	3.11	3.31	3.37	3.44	3.58
	MLC-Toolbox	10	0.74	0.02	0.72	0.73	0.74	0.75	0.78
	MULAN	10	3.81	0.18	3.48	3.80	3.88	3.92	3.96
	scikit-multilearn	10	2.96	0.05	2.86	2.92	2.98	2.99	3.01
tmc2007	MEKA	10	85.39	3.73	76.84	86.48	86.92	87.17	88.28
	MLC-Toolbox	10	139.91	0.29	139.52	139.66	139.93	140.10	140.33
	MULAN	10	89.85	3.21	80.90	90.44	90.71	90.83	92.52
	scikit-multilearn	10	32.36	1.50	29.61	31.72	33.18	33.32	33.68
yeast	MEKA	10	3.76	0.14	3.47	3.73	3.78	3.87	3.90
	MLC-Toolbox	10	1.18	0.02	1.14	1.17	1.18	1.19	1.21
	MULAN	10	4.02	0.11	3.79	3.98	4.05	4.10	4.14
	scikit-multilearn	10	2.11	0.10	1.93	2.06	2.08	2.16	2.27

Table 4: User time of library running Label Powerset for each of the datasets.

set	library	count	mean	std	min	25%	50%	75%	max
Corel5k	MEKA	10	1666.84	1.71	1664.12	1665.87	1666.58	1668.02	1669.99
	MLC-Toolbox	10	1400.39	35.06	1336.14	1380.37	1396.89	1417.98	1462.22
	MULAN	10	1695.80	7.11	1687.45	1691.15	1694.73	1697.20	1712.86
	scikit-multilearn	10	1534.89	3.24	1530.85	1532.10	1534.55	1536.74	1540.01
bibtex	MEKA	10	1426.38	2.94	1422.25	1425.46	1426.30	1427.00	1431.46
	MLC-Toolbox	10	2852.94	165.46	2573.12	2791.99	2890.47	2929.04	3097.12
	MULAN	10	1432.34	2.81	1428.26	1430.74	1432.21	1433.66	1437.83
	scikit-multilearn	10	1333.13	2.86	1328.46	1331.33	1332.35	1335.19	1338.06
birds	MEKA	10	259.04	0.71	258.55	258.71	258.85	259.02	260.97
	MLC-Toolbox	10	65.78	0.81	64.73	65.28	65.50	66.16	67.57
	MULAN	10	258.67	1.59	256.11	257.51	258.94	259.77	261.12
	scikit-multilearn	10	58.05	0.05	57.97	58.02	58.04	58.07	58.12
delicious	MEKA	10	14385.92	7.52	14378.90	14381.37	14383.44	14385.24	14399.84
	MLC-Toolbox	10	5791.13	170.72	5492.74	5709.92	5826.34	5919.17	5987.11
	MULAN	10	14661.50	7.65	14645.94	14657.66	14661.54	14666.47	14673.41
	scikit-multilearn	10	14213.03	28.43	14172.85	14192.80	14211.53	14241.15	14246.31
emotions	MEKA	10	139.68	1.61	136.51	139.08	139.32	140.76	142.36
	MLC-Toolbox	10	63.48	0.72	62.96	63.22	63.22	63.42	65.48
	MULAN	10	162.92	1.69	159.86	162.52	163.02	163.83	165.68
	scikit-multilearn	10	51.07	0.05	50.99	51.05	51.08	51.10	51.17
enron	MEKA	10	707.42	2.56	701.80	707.04	707.94	708.44	711.58
	MLC-Toolbox	10	215.96	10.90	200.23	207.90	215.11	225.24	232.22
	MULAN	10	714.13	4.37	710.48	712.62	713.34	713.60	726.18
	scikit-multilearn	10	156.54	0.61	155.61	156.04	156.45	157.09	157.38
genbase	MEKA	10	649.53	0.66	648.36	649.28	649.60	649.76	650.84
	MLC-Toolbox	10	114.91	4.80	109.46	111.19	114.27	117.05	125.15
	MULAN	10	640.55	1.35	639.77	639.84	640.01	640.15	643.92
	scikit-multilearn	10	125.80	0.02	125.78	125.79	125.80	125.81	125.84
mediamill	MEKA	10	13629.82	6.41	13624.72	13625.81	13628.58	13630.40	13646.53
	MLC-Toolbox	10	2938.81	12.77	2922.50	2927.15	2939.71	2949.28	2955.52
	MULAN	10	13582.38	50.91	13547.02	13550.89	13553.20	13621.77	13674.80
	scikit-multilearn	10	12801.70	13.43	12773.04	12797.68	12802.40	12809.80	12821.79
medical	MEKA	10	490.32	4.03	487.45	487.55	489.15	489.94	498.78
	MLC-Toolbox	10	129.75	7.82	121.34	123.38	126.39	137.62	140.70
	MULAN	10	545.42	1.74	543.10	544.33	544.88	546.41	549.15
	scikit-multilearn	10	89.29	0.02	89.26	89.28	89.28	89.30	89.31
scene	MEKA	10	501.84	1.37	500.18	500.84	501.25	503.01	504.34
	MLC-Toolbox	10	100.44	1.48	98.37	99.48	100.06	101.22	103.34
	MULAN	10	619.23	13.63	592.73	621.83	624.34	625.81	630.77
	scikit-multilearn	10	106.02	0.01	106.00	106.01	106.02	106.02	106.04
tmc2007	MEKA	10	2873.22	1.60	2870.86	2871.82	2873.66	2874.12	2875.92
	MLC-Toolbox	10	1989.50	140.75	1767.65	1911.14	1975.03	2041.82	2284.26
	MULAN	10	2886.81	4.44	2882.71	2883.54	2885.84	2888.71	2897.22
	scikit-multilearn	10	2729.93	5.23	2719.09	2728.22	2731.22	2733.61	2735.87
yeast	MEKA	10	621.49	1.36	619.78	620.30	621.75	621.96	624.14
	MLC-Toolbox	10	130.58	1.29	129.18	129.71	130.50	130.96	133.66
	MULAN	10	629.12	1.26	626.62	628.23	629.89	629.94	629.98
	scikit-multilearn	10	84.67	0.16	84.45	84.54	84.67	84.79	84.89

Table 5: Maximum memory usage (in kilobytes) of library running Label Powerset for each of the datasets.