Journal of Machine Learning Research 0 (0000) 0-00 Submitted 0/00; Published 00/00

Multi-Label Classification Toolbox

Keigo Kimura KKIMURA@MAIN.IST.HOKUDAI.AC.JP
Department of Computer Science
Hokkaido University

Lu Sun SUNLU@MAIN.IST.HOKUDAI.AC.JP
Department of Computer Science
Hokkaido University

Mineichi Kudo MINE@MAIN.IST.HOKUDAI.AC.JP
Department of Computer Science
Hokkaido University

Editor: John Doe

Abstract

Multi-Label Classification toolbox is a MATLAB/OCTAVE library fol Multi-Label Classification (MLC). There exist a few JAVA libraries for MLC, but there is no MATLAB/OCTAVE library that covers various methods. This toolbox offers an environment for evaluation, comparison and visualization of the MLC results. One attraction of this toolbox is that it enables to try many combinations of feature space dimension reduction, sample clustering, label space dimension reduction and ensemble, etc.

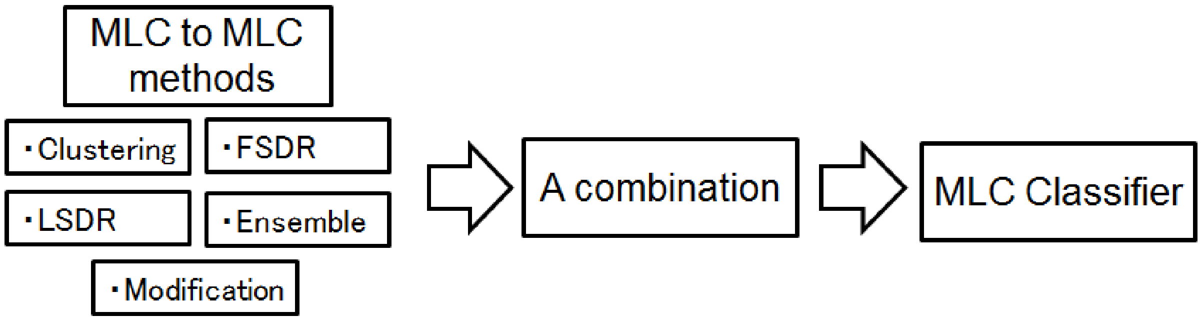
Keywords: Multi-Label Classification, Multi-Label Learning, MATLAB/OCTAVE

1. Multi-Label Classification and Libraries

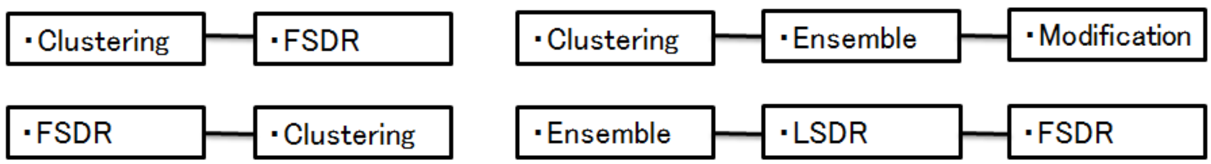
Multi-Label Classification (MLC), a problem which allows an instance to have more than one label at the same time, becomes popular since the problem fits real applications more than traditional single-label classification. Since combinations of labels must be considered to solve MLC, MLC gives many challenging problems and many works are trying to solve them.

There are already a few libraries for MLC available. Mulan is the most popular Java library for MLC developed by Tsoumakas et al. (2011). This library provides not only many MLC methods but also many MLC datasets. MEKA is another popular Java library for MLC developed by Read et al. (2016). MEKA provides GUI and is specialized for a family of methods called Classifier Chain (CC) (Read et al. (2011)). Both of Mulan and MEKA are based on Weka, that is, a popular classification Java library developed by Hall et al. (2009). Thus, both of Mulan and MEKA inherit methods implemented on Weka such as decision trees or SMO algorithm for SVMs. On the other hand, recent many implementations of MLC methods are based on MATLAB/OCTAVE. For example, as summarized on de Carvalho and Freitas (2009), LAMDA group[1] or Dr. ML-Zhang.[2] Indeed, MATLAB/OCTAVE implementations are faster than that in Java in general. This

1. `http://lamda.nju.edu.cn/Data.ashx`
2. He is also providing some Java codes on `http://cse.seu.edu.cn/people/zhangml/Resources.htm`



(a) An overview of MLC toolbox. MLC toolbox enable to hadnle many combinations of MLC methods



(b) Examples of combinations

Figure 1: An illustration of MLC toolbox.

is also helped by MEX technique taht incorporates with C++ codes. This is our motivation to have developed this library.

2. MLC toolbox

This toolbox, a MATLAB/OCTAVE library for MLC, offres easy implementation of various MLC methods and enables to compare them with each other fast and easily. There are many publicly available feature extraction methods and label space dimension reduction methods for MLC in MATLAB/OCTAVE codes. However, some clustering based methods or ensemble methods such as HOMER or RAkEL are not publicly available yet. This is probably because MATLAB/OCTAVE is specialized vector/matrix operations. This specialty might have produced two different group: the one is the group using Mulan or MEKA and the other using MATLAB/OCTAVE. This separation problem should be resolved to enable comparison over these two groups. MLC toolbos provides one solution for this goal. e different implementation environments makes the comparison complicated. Thus, only a few papers has compared methods transversally. MLC toolbox provides some methods not familiar with MATLAB/OCTAVE environments.

The most attractive point of MLC toolbox is that you can handle many combinations of MLC methods. As known well, many MLC methods convert a given MLC problem into more than one simpler or smaller MLC problems. For example, CBMLC (Nasierding et al. (2009)) use a clustering and produce smaller MLC problems. Feature extraction methods can also be considered as converting a problem into a smaller problem w.r.t. the size of features. These can be easily combined together in any order. For example, SLEEC (Bhatia et al. (2015)) combines a clustering and a feature extraction. These combinations exist in

Table 1: Comparison of Libraries

	Mulan	MEKA	LAMDA	ML-Zhang	**MLC-toolbox**
Launguage	Java	Java	MATLAB	MATLAB	MATLAB
	Available Methods				
Combination					✓
RSL	✓	✓			✓
BR	✓	✓			✓
LC	✓	✓			✓
PW	✓				✓
Dependencies	✓	✓	✓	✓	✓
NN	✓		✓	✓	✓
Deep NN	✓				
SVMs				✓	✓
Meta-Learners	✓	✓			✓
Imbalance				✓	✓
FSDR			✓	✓	✓
LSDR					✓
Clustering					✓
Thresholding	✓	✓			✓

myriad. MLC toolbox enables to try much more combinations of MLC methods in any orders more easily (See Fig.).

MLC toolbox is a library on MATLAB/OCTAVE and is flexible enough for allowing you to add your own method. Unfortunately, some dependencies exist. For example,it requires LIBLINEAR Fan et al. (2008) and LIBSVM Chang and Lin (2011) for fast computation. On the other hand, Mulan and MEKA depend only on Weka. Nevertheless, this toolbox is OS-agnostic and thus offhand.

MLC toolbox focuses only on supervised MLC. Thus, semi-supervised, incremental and missing labels are not supported. This is a drawback of this library against MEKA.

Table 1 summarizes the comparison of MLC methods available. The category of MLC methods are mostly based on de Carvalho and Freitas (2009) but with some modifications: Ranking via Single-Label learning (RSL), Binary Relevance (BR), Label-Combination (LC), Pair-Wise comparison (PW), Dependencies, Neural-Network (NN), Deep Neural Network (DNN), Support Vector Machines (SVMs), Meta-Learners, Imbalance, Feature Space Dimension Reduction (FSDR), Label Space Dimension Reduction (LSDR), Clustering-based method (Clustering), Thresholding strategies and algorithms (Thresholding)

3. How to Use MLC toolbox

3.1 Set up

MLC toolbox needs MATLAB and statistical and machine learning toolbox provided by Mathworks. After downloading MLC toolbox, MLC toolbox requires only to compile some mex functions by running compileMEXfunctions.m on the main directory folder. Once you add the path to the all folders in MLC toolbox, you can run every program of MLC toolbox.

3.2 Running Sample codes

As an example, we describe how to run Sample.m (a sample code to run MLC methods) briefly.
Dataset:

```
datasetname='scene';  ## set the dataset name
numCV=5;  ## numCV=3 or numCV=10
```

Method:

```
method.name={'PCA','CBMLC','rCC'};
```

As we stated above, MLC toolbox can handle combinations of MLC methods. MLC toolbox combine components in this order.
Parameters: This library does no provide an efficient way to set parameters. This is because it is mostly impossible to set parameters on myriad combinations of methods. Thus, this library needs parameter files to set parameters for each MLC method. For sample codes, as an example, we give a parameters file (setMethodParameter.m) for each method. For example, to change parameter of PCA, change values of SetPCAParameter.m:

```
function[param]=SetPCAParameter(~)
param.dim=300;  or param.dim='numF*0.5';  ## numF is a number of features
```

The latter definition refers to the dataset information (reducing 50% feature dimensions).
Base classifier: Most of all methods except algorithm adopting methods calls traditional binary/multi-class classifiers to solve MLC. We call base classifiers by

```
method.base.name='linear_svm';  ## 'ridge','svm','knn' are available
method.base.param.svmparam='-S 2 -q' ##paramters
```

Thresholding: threshold method and parameters to obtain discrete results by

```
method.th.type='Scut';  ## 'Rcut','Pcut' are also available
method.th.param=0.5; ##paramters
```

See methods on Tang et al. (2009).

3.3 Implementing your code in MLC-toolbox

Users do not have to understand, the whole part of MLC toolbox to extend implement your own codes. For implementing a new method called Newmethod, MLC toolbox requires to implement, Newmethod_train.m for training, Newmethod_test.m for testing and SetNewmethodParameter.m for parameter setting (not necessary but recommended). More detailed instruction information is given on a tutorial document.

4. Summary

This MLC toolbox is the first toolbox that contains the largest set of methods and these combinations for MLC on MATLAB/OCTAVE. The MLC toolbox can be available free for academic purposes. The software, documents, and tutorial slides can be available at `https://github.com/KKimura360/MLC_toolbox`. We welcome your feedback and contribution to this toolbox.

References

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 28*, pages 730–738. 2015.

Chih-Chung Chang and Chih-Jen Lin. Libsvm: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(3):27, 2011.

André CPLF de Carvalho and Alex A Freitas. A tutorial on multi-label classification techniques. In *Foundations of Computational Intelligence Volume 5*, pages 177–195. Springer, 2009.

Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9 (Aug):1871–1874, 2008.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. The weka data mining software: an update. *ACM SIGKDD explorations newsletter*, 11(1):10–18, 2009.

Gulisong Nasierding, Grigorios Tsoumakas, and Abbas Z Kouzani. Clustering based multi-label classification for image annotation and retrieval. In *IEEE International Conference on Systems, Man and Cybernetics, 2009. SMC 2009.*, pages 4514–4519. IEEE, 2009.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333, 2011.

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.

Lei Tang, Suju Rajan, and Vijay K Narayanan. Large scale multi-label classification via metalabeler. In *Proceedings of the 18th international conference on World wide web*, pages 211–220. ACM, 2009.

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