

MEKA: A Multi-label/Multi-target Extension to WEKA

Jesse Read

*Helsinki Institute for Information Technology (HIIT), and
Aalto University, Dept. of Computer Science, Espoo, Finland*

Peter Reutemann

Bernhard Pfahringer

Geoff Holmes

*Department of Computer Science
University of Waikato, New Zealand*

JESSE.READ@AALTO.FI

FRACPETE@WAIKATO.AC.NZ

BERNHARD@CS.WAIKATO.AC.NZ

GEOFF@CS.WAIKATO.AC.NZ

Editor: Balazs Kegl

Abstract

Multi-label classification has rapidly attracted interest in the machine learning literature, and there are now a large number and considerable variety of methods for this type of learning. We present MEKA: an open-source Java framework based on the well-known WEKA library. MEKA provides interfaces to facilitate practical application, and a wealth of multi-label classifiers, evaluation metrics, and tools for multi-label experiments and development. It supports multi-label and multi-target data, including in incremental and semi-supervised contexts.

Keywords: classification, learning, multi-label, multi-target, incremental

1. Introduction

In *multi-label learning* a data instance may be associated with multiple binary class labels. This is as opposed to the traditional task of single-label (i.e., multi-class, or binary) classification where each instance is only associated with a single class label. The multi-label context is receiving a lot of attention and is relevant to a wide variety of domains, including text, music, images and video, and bioinformatics; as reviewed by Tsoumakas et al. (2010).

The multi-label learning problem is in fact a special case of *multi-target* learning (also known as multi-dimensional or multi-objective), where each label can take multiple values, as opposed to binary labels indicating relevance or not (Bielza et al., 2011; Appice and Džeroski, 2007).

The context of multiple target variables has important implications for classification (how to model dependencies between variables) and evaluation (how to score multiple target classifications for each instance) that traditional single-label frameworks do not deal with. MEKA has been designed specifically for this context.

2. The MEKA Framework

MEKA was created to perform and evaluate multi-label classification using the popular and effective family of *problem transformation* methods, which make use of existing off-the-shelf

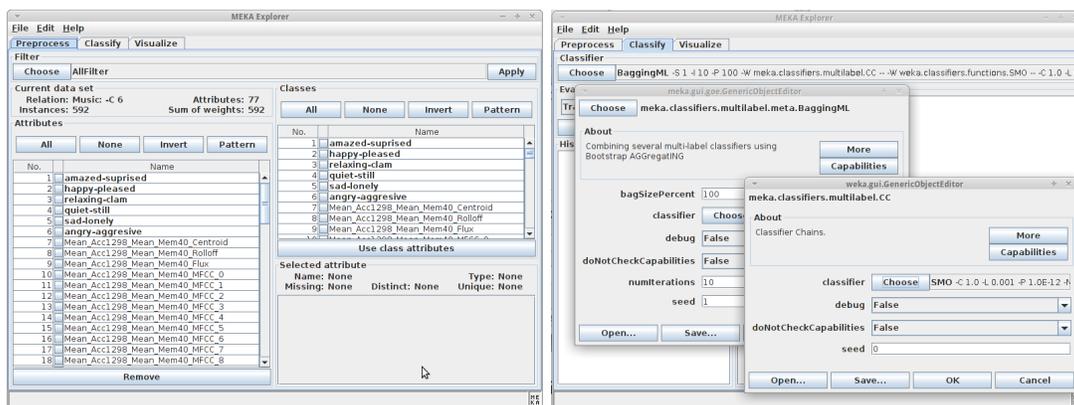


Figure 1: MEKA’s GUI interface; having loaded and setup the Music data set (left) and selecting and configuring a classifier (right).

single-label (binary or multi-class) methods as ‘base classifiers’. The WEKA framework (Hall et al., 2009) contains a plethora of well-documented single-label classifiers used by the machine learning community for many years that can be used as such.

MEKA contains all the basic problem transformation methods as reviewed by Tsoumakas et al. (2010), advanced methods including many of those surveyed by Madjarov et al. (2012) and Zhang and Zhou (2013), many varieties of classifier chains (Read et al., 2011) which have often been used as a benchmark in the recent multi-label literature, and also algorithm adaptations such as multi-label neural networks and deep neural networks (Hinton and Salakhutdinov, 2006). It includes two strategies for automatic threshold calibration, and a variety of evaluation metrics from the literature.

MEKA is easily used from either the command line interface (CLI) or graphical user interface (GUI). Thus no programming is required to parameterize, run, and evaluate classifiers, making it suitable for practitioners unfamiliar with Java. However, it is straightforward to extend MEKA with new classifiers and integrate it in other frameworks. Those familiar with WEKA will have almost no learning curve (much of WEKA’s documentation and modus operandi is directly applicable). Any new MEKA classifier can also be combined within any of MEKA’s existing ensemble schemes and any WEKA base classifier without writing extra code, and be compared easily with benchmark and state-of-the-art methods. MEKA also offers support for semi-supervised and incremental classification in the multi-label context (Read et al., 2015), as well as the general multi-target context considered by Appice and Džeroski (2007), making it a good platform for working in these areas.

2.1 Using MEKA

MEKA can be run on any machine with Java installed (version 1.7 or above). Everything needed to run MEKA is included in the distribution package and detailed instructions and examples on how to setup data sets and run experiments are included in the tutorial. Figure 1 shows screen captures of MEKA’s GUI.

MEKA uses WEKA’s thoroughly-documented ARFF file format,¹ using multiple binary attributes to specify the label relevances (or nominal attributes in the case of multi-target). The target attributes can easily be configured using the GUI, or directly from within the ARFF file (optionally using WEKA’s filters).

MEKA’s CLI follows the style of WEKA. For example, to run five fold cross validation of an ensemble of 50 chain classifiers (Read et al., 2011) on the Music data set with support vector machines as the base classifier (as also shown on the right of Figure 1):

```
java meka.classifiers.multilabel.meta.BaggingML -x 5 -t data/Music.arff -I 50 \
-W meka.classifiers.multilabel.CC -- -W weka.classifiers.trees.SMO
```

A list of methods implemented in MEKA along with examples, is given at <http://meka.sourceforge.net/methods.html>. Further examples and documentation can be found in the tutorial and API reference, where relevant publications are also mentioned. MEKA also includes a wrapper to MULAN (the `MULAN` class) to allow additional classifiers to be run from MEKA’s interfaces. MEKA includes over a dozen evaluation metrics, including Hamming loss, 0/1 loss, Jaccard index, rank loss, log loss, and F1-measure (macro and micro averages). Several metrics can be output overall and per-label.

2.2 Extending MEKA

MEKA can be extended by writing classifiers that implement the `MultiLabelClassifier` interface. A multi-label classifier uses the same methods as a WEKA classifier, namely the `buildClassifier(Instances)` and `distributionForInstance(Instance)` methods. The difference in MEKA is that `classIndex()` will indicate the *number* of target attributes. As in WEKA, the `distributionForInstance` method returns a vector which may be votes or posterior probabilities. In the latter case, a threshold or thresholds can automatically be calibrated, or defined ad-hoc according to user preference, to produce 0/1 label relevances. The tutorial deals with the general multi-target case.

Incremental classifiers simply implement `IncrementalMultiLabelClassifier` and thus, as with WEKA’s `UpdateableClassifier` interface, have an `updateClassifier(Instance)` method. MEKA automatically carries out incremental learning and evaluation for any updateable classifiers. Semi-supervised classifiers implement `SemisupervisedClassifier` interface and must implement the method `setUnlabelledData(Instances)` which is called by the evaluation routine.

3. Related frameworks

MULAN (Tsoumakas et al., 2011) is a programmatic API also based on WEKA providing a number of multi-label algorithms implemented from the literature. There is some overlap in terms of classifiers available, but the usage and performance is different. MULAN is an API, whereas MEKA provides both graphical and command-line interfaces and this allows novel classifier and ensemble combinations and parameterizations thereof to be trialled rapidly without writing any code; allowing much flexibility when tackling a multi-label problem. A single command line is often sufficient to reproduce the results of publications. Both frameworks have attracted considerable interest and both are mentioned in the literature.

1. See, for example, <http://weka.wikispaces.com/ARFF>.

	CLUS	MULAN	MEKA
interface	config. file		CLI, GUI
multi-target	✓	regression	classification
hierarchical	✓	✓	✓
incremental	✓		✓
semi-supervised	✓		✓
num. of classifiers	2+	20+	20+
classification paradigms	decision trees/rules	WEKA+others	WEKA+others

Table 1: Comparing MEKA with related frameworks.

Data set	MEKA	MULAN
Enron	8	33
Corel-5k	10	35
MediaMill	449	2104

Table 2: Running time (s) of RAKEL under MEKA and MULAN, with SMO, $k = 3, m = 10$.

MEKA specializes in the problem transformation methods, in particular the classifier chains paradigm, implementing at least 10 variations; and also meta methods for combining them together in many combinations.

CLUS is a decision tree and rule learning system (Appice and Džeroski, 2007) that can also carry out multi-label and multi-target classification. Classifiers are configured in a text file. CLUS’s implementations of decision tree and rule algorithms are scalable and competitive; but it does not include many of the popular problem transformation methods.

Table 1 roughly compares the frameworks in terms of features. CLUS has fewer classification paradigms, but can be used in many different contexts. MULAN has a similar number of classifiers but MEKA has different methods. MEKA is noticeably faster than MULAN in some implementations under the same configurations, for example RAKEL, see Table 2.

MEKA itself integrates well in other frameworks. For example, it is already part of the MOA (data streams), DKPRO (text classification), and ADAMS (workflow engine) frameworks, and available via Maven Central.²

4. Summary and Further Notes

MEKA provides a multi-label and multi-target framework with comfortable command line and graphical interfaces. This includes traditional, benchmark, and state of the art algorithms; and a multitude of multi-label specific evaluation measures. It also includes methods for incremental, multi-target, and semi-supervised learning, and is easy to use and extend. MEKA is released under the GNU GPL 3 licence; can run on any Java (1.7 or above) machine. The software with source code, API reference, data sets, list of methods with examples, and a tutorial can be found at <http://meka.sourceforge.net>, along with links to additional material about learning from multi-label and multi-target data.

2. See <http://moa.cms.waikato.ac.nz/>, <https://code.google.com/p/dkpro-tc/>, <https://adams.cms.waikato.ac.nz/>, and <https://search.maven.org>, respectively

References

- Annalisa Appice and Saso Džeroski. Stepwise induction of multi-target model trees. In *ECML '07: Proceedings of the 18th European Conference on Machine Learning*, pages 502–509, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-74957-8. doi: 10.1007/978-3-540-74958-5_46. URL http://dx.doi.org/10.1007/978-3-540-74958-5_46.
- Concha Bielza, Guangdi Li, and Pedro Larrañaga. Multi-dimensional classification with bayesian networks. *International Journal of Approximate Reasoning*, 52:705–727, 2011.
- Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Reutemann Peter, and Ian H. Witten. The weka data mining software: An update. *SIGKDD Explorations*, 11(1), 2009.
- Geoffrey Hinton and Ruslan Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504 – 507, 2006.
- Gjorgji Madjarov, Dragi Kocev, Dejan Gjorgjevikj, and Saso Dzeroski. An extensive experimental comparison of methods for multi-label learning. *Pattern Recogn.*, 45(9):3084–3104, 2012. ISSN 0031-3203. doi: 10.1016/j.patcog.2012.03.004. URL <http://dx.doi.org/10.1016/j.patcog.2012.03.004>.
- Jesse Read, Bernhard Pfahringer, Geoffrey Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine Learning*, 85(3):333–359, 2011.
- Jesse Read, Albert Bifet, and Fernando Perez-Cruz. Deep learning in partially-labelled data streams. In *SAC 2015: 30th ACM Symposium on Applied Computing*. ACM, 2015.
- Grigorios Tsoumakas, Ioannis Katakis, and Ioannis P. Vlahavas. Mining multi-label data. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*. 2nd edition, Springer, 2010.
- Grigorios Tsoumakas, Jozef Vilcek, Eleftherios Spyromitros Xioufis, and Ioannis Vlahavas. MULAN: A java library for multi-label learning. *Journal of Machine Learning Research*, 12:2411–2414, 2011.
- Min-Ling Zhang and Zhi-Hua Zhou. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, 99(PrePrints):1, 2013. ISSN 1041-4347. doi: <http://doi.ieeecomputersociety.org/10.1109/TKDE.2013.39>.