Multi-label Learning with MEKA

Vaishali S. Tidake
Research Scholar, Dept. of Computer Engineering
NDMVPS's KBTCOE, Nashik

Shirish S. Sane
Vice Principal and Head, Dept. of Computer Engineering
KKWIEER, Nashik

I. Introduction to Multi-label Learning

Consider a photograph of sky, second photograph of forest, and third photograph having both sky and forest. If you want to label these photographs by their contents, then first and second photographs can be labeled straight away as sky and forest. But the third photograph has to be labeled by both sky and forest. This scenario represents the multi-label learning where a photograph may have multiple labels.

Classification is a commonly used data mining task. It uses supervised learning in which a model is trained from a set of known instances, called train set. Each instance in a train set has a set of values one each for the fix number of descriptive features/attributes and a pre-defined class label. Thus each of the training instances belongs precisely to only one class. Once the model is trained and tested, it is used to classify unseen instances. This is known as Single label (SL) classification and has been used in several distinct domains. However, in certain domains such as text categorization (TC), annotation of image, audio and video, bioinformatics, emotion recognition systems, etc, where instances may belong to one or more classes, the SL classification techniques cannot be used. The set of techniques that can handle instances having multiple labels has been developed and are called multi-label (ML) classification. In older days multi-label classification/learning was only used for categorization of text. But it has been used in the recent past for discovery of drug, tag recommendation, prediction of gene function [1] etc. Therefore it has become an upcoming research field in the area of machine learning. This article focuses on MEKA which is a tool designed for Multi-label learning.

II. Basics of MEKA

A. Introduction

MEKA is an open source framework which supports multi-label learning [4]. MEKA uses WEKA software as its base that supports SL classifiers only [3]. MEKA provides framework for machine learning. It helps to develop, run and evaluate various multi-label classifiers.

B. Installing and Using the tool

The latest version of MEKA 1.9.0 is released on 04th Nov 2015. It is available for download as the *MEKA-release-1.9.0.zip* file from the MEKA website directly [4]. This file should be extracted to create the *MEKA-1.9.0* directory which contains necessary APIs, few samples of multi-label datasets, packages and files to run the software. The tool can be used in two ways, either from command line or using GUI. To open the MEKA GUI, use *run.bat* for Microsoft Windows (*./run.sh* for Linux). Fig. 1 shows the MEKA GUIChooser screen as the first screen after opening MEKA. The GUI shows two options "Explorer" and "Experimenter".

Fig. 1 : MEKA GUIChooser

This article deals only with "Explorer". When someone clicks on "Explorer" button it shows different menus and tabs on the screen as shown in Fig. 2.

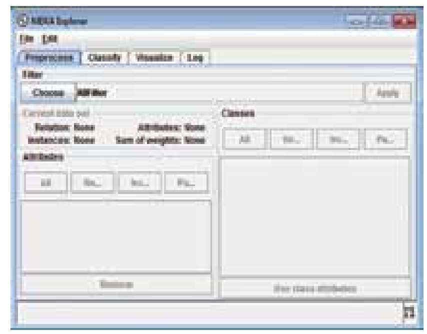

Fig. 2 : MEKA Explorer

C. Loading the Input Data

To perform any operation, it is necessary to load the required data. MEKA needs data in ARFF format which is same as that followed by WEKA files.

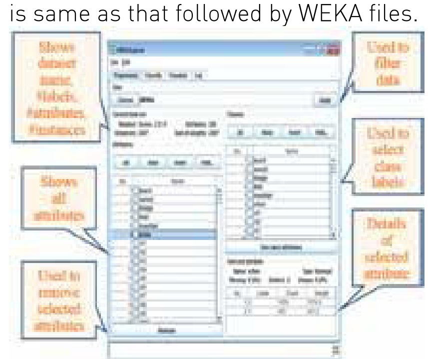

Fig. 3 : MEKA Explorer showing loaded dataset

In MEKA Explorer, the input dataset can be loaded using *File* menu with *Open* option. The loaded dataset can be viewed in the *Preprocess* tab as shown in Fig. 3. MEKA Explorer shows various information of the loaded dataset in the preprocess tab. On the left side of window, it has a "filter selection" option, dataset and attribute information, and at the bottom is the "remove" button to be used for removal of unwanted attributes, if any. On the right side, there is option to choose and set class labels among the attributes, followed by selected attribute section showing name, data type and related information of an attribute selected in the left section.

D. Multi-label Classification

Several different methods have been developed and reported in the literature to perform multi-label learning task. Two broad categories used to perform multi-label learning are the *problem transformation* and the *algorithm adaptation* [6].

The *problem transformation* approach involves transformation of an input instance into a representation suitable for traditional single-label classifier. In this approach, the multi-label data representation is transformed into a single-label data representation which is acceptable by traditional SL classification methods. In simple words, problem transformation operates on the principle *"fit data to algorithm"* [7]. Different algorithms which come under this approach are BR, LP, CC, RPC, CLR, etc.

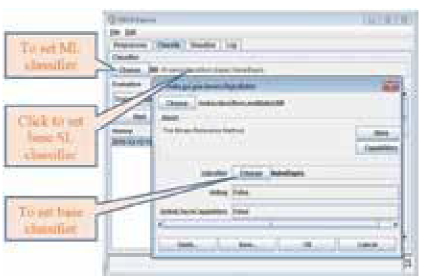

Fig. 4 : MEKA Explorer showing settings of BR classifier

The *algorithm adaptation* approach on the other hand, involves modification of an existing SL classifier algorithm making it suitable to handle multi-label instances [7], [8]. In simple words, algorithm adaptation operates on the principle *"fit algorithm to data"* [7]. Many algorithms such as MLkNN, ML-BPNN, ML-DT, etc. follow this approach.

Ensemble method is also considered an important approach used for multi-label learning that combines outcomes from several classifiers based on either problem transformation or algorithm adaptation and has provided better results [2]. Algorithms such as 'RAkEL', Ensembles of classifier chains (ECC), etc. follow this approach.

As MEKA is designed to support multi-label classification, let us see how to perform it using MEKA. This option is provided in *Classify* tab as shown in Fig. 4.

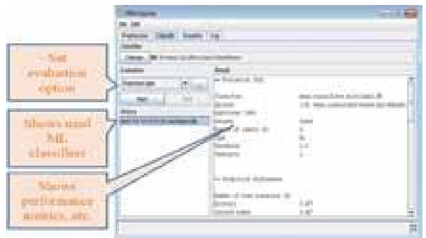

Fig. 5 : MEKA Explorer showing Result of BR classifier

- **Using the Binary Relevance (BR) Classifier**

The BR classifier is a widely used method that takes problem transformation approach. In this method, a multi-label problem is converted into |L| number of binary SL classification problems where L is a set of labels. Each of the binary classifiers votes separately to get the final result [6], [9].

As shown in Fig. 4, the *Classifier* option in *Classify* tab provides variety of ML classifiers. If BR is selected, one needs to specify the base SL classifier by clicking on "BR" that pop ups an option window called *GenericObjectEditor* to choose desired SL classifier. 'NaiveBayes' SL classifier has been selected in Fig. 4. If not selected, default classifier J48 is used.

After selecting a ML classifier, an *Evaluation* option is to be set like Train/test split, Cross-validation, etc. as shown in Fig. 5. If not selected, the first option Train/test split is used for Evaluation. Clicking the *Start* button executes the selected ML classifier on the selected ML dataset and provides *Results* as shown on the right side in the MEKA Explorer in Fig. 5. Results include *evaluation information* and *predictive performance*. If multiple classifiers are run one by one, then their list is displayed in *History* section of MEKA Explorer.

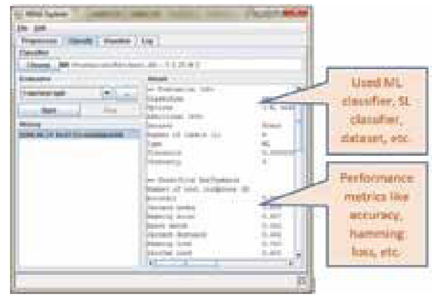

Fig. 6 : MEKA Explorer showing various performance metrics as a result

The *evaluation information* portion in *Results* section shows ML classifier used, base SL classifier used, dataset used, its type and related information. The *predictive performance* portion in *Results* section shows values of various performance metrics like accuracy, hamming loss, one error, etc. Accuracy is related to the labels which are predicted correct. Hamming loss is related to the misclassification of an instance and label pairs. One error is useful to measure when the generated top ranked label does not belong to relevant labels of instances.

- **Label Powerset (LP)**

BR does not consider relationship between labels. This drawback of BR is overcome by LP, also called as LC (Label Cardinality). In LP classifier, each distinct combination of labels is considered as a separate class and the entire problem is treated as a multi-class single-label (MSL) problem [2]. LP classifier also requires use of base classifier like J48 as shown in Fig. 7. However, it performs poorly when there is what is called 'class imbalance'.

Fig. 7 : MEKA Explorer showing LP classifier

- **Classifier Chain (CC)**

The Classifier Chain (CC) approach [2], [12], like LP, also try to overcome the drawback of BR. Similar to BR, a ML problem is transformed into |L| number of SL problems where L denotes a set of labels and for each label L_j, a separate binary classifier C_j is designed. But the input for each classifier C_j is different. Like LP classifier, CC also needs selection of base classifier. If not set, by default J48 SL classifier is used as shown in Fig. 8.

Fig. 8 MEKA Explorer showing CC classifier

- **Multi-label Back Propagation Neural Network (ML-BPNN)**

In Back Propagation NN, once the output is generated and if it is different than the desired output, then an error is calculated and it is used to make changes the weights in the previous layers. This operation is performed for each instance in the training set and for each label in that instance, calculated errors are added up. Addition of all such errors for all the instances is computed and is minimized to improve the performance using the correlation between labels of all the instances that belong to particular instance and labels but do not belong to that instance [13].

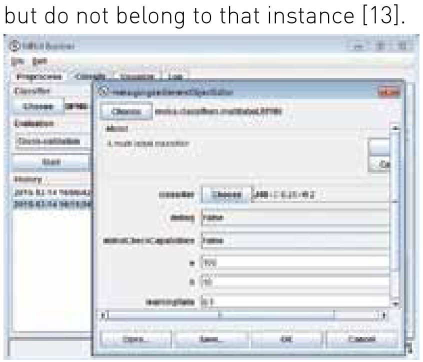

Fig. 9 MEKA Explorer showing ML-BPNN classifier

As shown in Fig. 8, ML classifier BPNN can be selected from *Classifier* option in the *Classify* tab. One needs to set base SL classifier. Otherwise default J48 classifier is used as shown in Fig. 9.

- **Random k-Labelsets (RAkEL)**

The problem of class imbalance in LP is removed in Random k-Labelsets. RAkEL is actually an *ensemble* of multiple LP classifiers. It combines various LP classifiers having different k combinations of all labels referred as *labelsets* [2], [11].

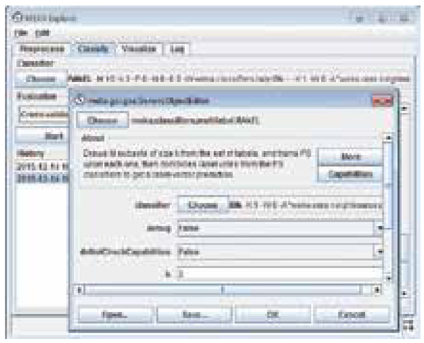

Fig. 10 MEKA Explorer showing RAkEL classifier

Table 1 describes three multilabel datasets namely music, yeast and enron. Table 2, 3 and 4 show comparison of few ML classifiers for three parameters accuracy, hamming Loss and one error respectively using MEKA 1.9.0. Always larger value is expected for accuracy and smaller value is expected for hamming loss and one error. Tables show that BPNN has given better results among all classifiers for the given three datasets on three parameters used.

Table 1 : Datasets used

Dataset	No. of Records	No. of Attributes	No. of Labels
music	592	77	6
yeast	2417	117	14
enron	1702	1054	53

Table 2 : Comparison of various classifiers for Accuracy

Dataset	BR	LP	CC	BPNN	RAkEL
music	0.39	0.445	0.408	**0.546**	0.523
yeast	0.435	0.403	0.413	**0.521**	0.416
enron	0.388	0.355	**0.414**	0.347	0.027
Avg	0.404	0.401	0.411	**0.471**	0.322

Table 3 : Comparison of various classifiers for Hamming Loss

Dataset	BR	LP	CC	BPNN	RAkEL
music	0.318	0.281	0.295	**0.21**	0.249
yeast	0.256	0.278	0.278	**0.213**	0.325
enron	0.06	0.068	**0.054**	0.066	0.065
Avg	0.211	0.209	0.209	**0.163**	0.213

Table 4 : Comparison of various classifiers for One error

Dataset	BR	LP	CC	BPNN	RAkEL
music	0.49	0.48	0.5	**0.262**	0.297
yeast	0.481	0.541	0.522	**0.245**	0.42
enron	0.387	0.463	0.37	**0.359**	0.907
Avg	0.463	0.494	0.464	**0.288**	0.541

E. Prepreocessing

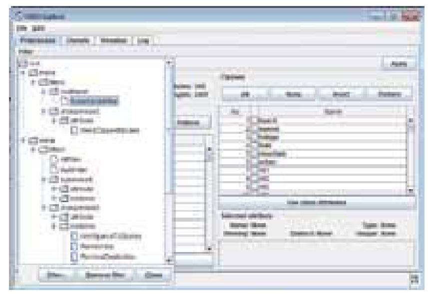

Fig. 11(a) MEKA Explorer showing filter selection

MEKA provides all the filters which are available in WEKA. These filters in WEKA are categorized as unsupervised and supervised. Each of these categories is further grouped into attribute related and instance related methods. MEKA, in addition to these filters, also provides few more filters that are not available in WEKA. As shown in Fig. 3, *Filter* section in *Preprocess* tab is used to select a desired filter. By clicking *Choose* button in the *Filter* section, various available filters are presented. Effect of selected filter on the dataset could be observed by clicking *Apply* button in the *Filter* section as shown in Fig. 3.

MEKA provides two more filters other than WEKA. One is multi-label *SuperNodeFilter* as shown in Fig. 11(a) and the other is unsupervised *MekaClassAttributes* filter which works at attribute level as shown in Fig. 11(b).

The *SuperNodeFilter* works with multi-label datasets. It helps the user to create super labels from the existing labels if the user has knowledge about existing labels. User needs to specify which labels should be grouped to form new super label that represents all the labels in that group. For example, if $\{L_1, L_2, L_3, L_4, L_5, L_6\}$ is the set of labels in the existing dataset such that L_1, L_3, L_4 form a group L_{134}. Also L_2, L_5 form L_{25} and L_6 forms a separate group. Then new dataset with set of constructed labels will be $\{L_{134}, L_{25}, L_6\}$. But this may sometimes cause changes in the values of the attributes.

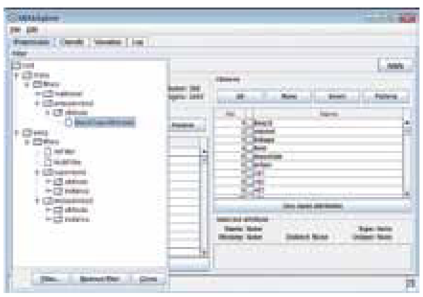

Fig. 11(b) : MEKA Explorer showing filter selection

Sometimes a need may arise to alter the sequence in which attributes are arranged or to change the class attributes in the dataset. MEKA assumes all the class attributes being placed at the beginning of the instance in the dataset. This can be achieved using an unsupervised filter operating at the attribute level provided in MEKA called *MekaClassAttributes* which is not available in WEKA, and is as shown in Fig. 11(b). User needs to select the attributes for reordering by specifying their numbers like 10 or a range like 7-10 as shown in Fig. 11(c).

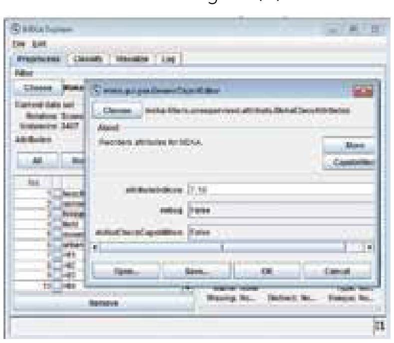

Fig. 11(c) : MEKA Explorer showing filter selection

After applying these settings, specified attributes will be placed at the beginning of the dataset as shown in Fig. 11(d). One can set *Use Class attributes* option provided in MEKA Explorer to change the number of class attributes. This number will be used by the classifier. However, if these attributes are numeric, then algorithm like BR cannot classify. So *Discretize* filter provided in WEKA filter list can be used first to convert numeric attributes to nominal type and then classification can be done.

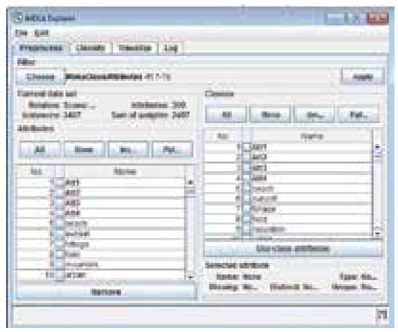

Fig. 11(d) MEKA Explorer showing filter selection

F. Datasets

MEKA accepts input data as an ARFF (Attribute Relation File Format). For example, scene.arff is a sample dataset which contains information about multimedia and is used in the literature for classification of scenes [2]. The dataset associates scenes to six different contexts such as beach, urban, mountain, field, sunset, foliage. MEKA versions come with few sample datasets. Other multi-label datasets are available for download from the website directly [4].

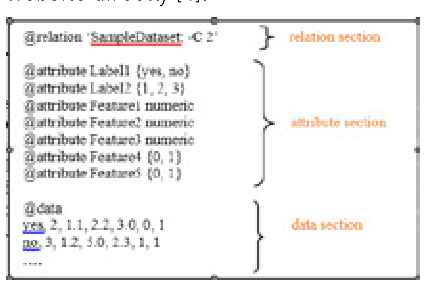

Fig. 12 : Sample multi-label dataset in ARFF format

ARFF represents Attribute Relation File Format. A sample ARFF file having two labels and five features is described in Fig. 12. Every ARFF file has two sections namely the header and the data. The header section contains the relation and the attribute section. The *relation* section is the first one which describes name of dataset and the number of labels in the dataset is specified after -C option. The *attribute* section is the second section where each line describes name of label/ feature and type of data that can be stored in it or a set of values which can appear for that attribute in case of nominal attribute. The *data* section is the third section which shows the data instances appeared in the dataset. Note that MEKA requires class labels to be preceded by features.

To summarize, MEKA is an excellent tool to perform ML classification on ML datasets. One can load a dataset, choose appropriate Classifier, apply attribute and Instance filters, if desired and choose desire evaluation option. It's an excellent tool for the researchers to carry out experiments to evaluate different classifiers. Explorer is suitable for beginners, 'Experimenter' tab provides advanced facilities. Use of some ML Algorithms such as BR, LP, RAkEL, CC, and ML-BPNN has been discussed briefly in this article.

References

[1] G. Tsoumakas, M. L. Zhang, Zhi-Hua Zhou, "Introduction to the special issue on learning from multi-label data", Mach Learn (2012) 88:1–4, DOI 10.1007/s10994-012-5292-9.

[2] G. Madjarov, D. Kocev, D. Gjorgjevikj, and S. Džeroski, "An extensive experimental comparison of methods for multi-label learning," Pattern Recognit., vol. 45, no. 9, pp. 3084–3104, 2012.

[3] M. Hall et al., "The WEKA data mining software: An update," SIGKDD Explor., vol. 11, no. 1, pp. 10–18, 2009.

[4] http://MEKA.sourceforge.net

[5] Mark Hall, Eibe Frank, Geo rey Holmes, Bernhard Pfahringer, Reutemann Peter, and Ian H. Witten. "The WEKA data mining software: An update", SIGKDD Explorations, 11(1), 2009.

[6] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," Int. J. Data Warehousing Mining, vol. 3, no. 3, pp. 1–13, 2007.

[7] M.L. Zhang and Z.H. Zhou, "A review on multi-label learning algorithms," , IEEE Transactions On Knowledge And Data Engineering, Vol. 26, No. 8, August 2014.

[8] A. de Carvalho and A. A. Freitas, "A tutorial on multi-label classification techniques," in Studies in Computational Intelligence 205, A. Abraham, A. E. Hassanien, and V. Snásel, Eds. Berlin, Germany: Springer, 2009, pp. 177–195.

[9] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multilabel data,", Data Mining and Knowledge Discovery Handbook, O. Maimon and L. Rokach, Eds. Berlin, Germany: Springer, 2010, pp. 667–686.

[10] G. Tsoumakas, M.-L. Zhang, and Z.-H. Zhou, "Tutorial on learning from multi-label data," in ECML PKDD, Bled, Slovenia, 2009 [Online]. Available:http://www.ecmlpkdd2009.net/wpcontent/uploads/2009/08/learning-from-multi-label-data.pdf.

[11] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multilabel classification," IEEE Trans. Knowl. Data Eng., vol. 23, no. 7, pp. 1079–1089, Jul. 2011.

[12] J. Read, B. Pfahringer, G. Holmes, E. Frank, "Classifier chains for multi-label classification", in: Proceedings of the 20th European Conference on Machine Learning, 2009, pp. 254–269

[13] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," IEEE Trans. Knowl. Data Eng., vol. 18, no. 10, pp. 1338–1351, Oct 2006.

About the Authors

Ms. Vaishali S. Tidake [CSI-I1504328] is currently working in Department of Computer Engineering at NDMVPS's KBTCOE, Nashik. She can be reached at vaishalitidake@yahoo.co.in.

Prof. Shirish S. Sane [CSI - 00008480] is currently working as Vice Principal and Head of the Computer Engineering Department at KKWIEER, Nashik. He is the Past Chairman of BOS in Computer Engineering, University of Pune. Currently he is working as Regional Vice President for CSI Region VI (Maharashtra & Goa). He can be reached at sssane@kkwagh.edu.in.

Call for Volunteers to Represent CSI in Technical Committees of IFIP

International Federation for Information Processing IFIP is the leading multinational, apolitical organization in Information & Communications Technologies and Sciences. Recognized by United Nations and other world bodies it represents IT Societies from 56 countries/regions, covering five continents with a total membership of over half a million. It links more than 3500 scientists from Academia & Industry, has over 100 Working Groups and 14 Technical Committees. Computer Society of India has been a Member of IFIP for long and has representation in all the Technical Committees. **We are looking for Members to represent CSI in the following Technical Committees of IFIP.**

- **TC-1 Foundation of Computer Science**
- **TC-3 IT in Education**
- **TC-6 Communication System**
- **TC-8 Information Systems**
- **TC-10 Computer Systems Technology**
- **TC-12 Artificial Intelligence**
- **TC-14 Entertainment Computing**
- **TC-2 Software: Theory and Practice**
- **TC-5 Information Technology Application**
- **TC-7 System Modelling and Optimization**
- **TC-9 Relationship between Computers & Society**
- **TC-11 Security & Protection on IP Systems**
- **TC-13 Human Computing Interaction**

Members interested in serving as TC Representative must be ready to attend the IFIP meetings at their own cost.
Members may please forward their profiles, with CSI Membership Number, organization details, list of past achievements in carrying out similar activities mentioning the TC of interest to Prof. Dr. Anirban Basu at president@csi-india.org and to Prof. Dr. A. K. Nayak at secretary@csi-india.org with a copy to sonali@csi-india.org before August 12, 2016. The email should have subject line: Interested in IFIP TC (Specify the number).

Dr. Anirban Basu
President, CSI

Prof. A K Nayak
Honorary Secretary , CSI