Evaluation of Multi-label Classifiers in Various Domains Using Decision Tree

Chapter · January 2018

DOI: 10.1007/978-981-10-7245-1_13

2 authors, including:

Vaishali S Tidake
Nashik District Maratha Vidya Prasarak Samaj's K.B.T. College of Engineering
8 PUBLICATIONS **9** CITATIONS

Evaluation of Multi-label Classifiers in Various Domains Using Decision Tree

V. S. Tidake and S. S. Sane

Abstract One of the commonly used tasks in mining is classification, which can be performed using supervised learning approach. Because of digitization, lot of documents are available which need proper organization, termed as text categorization. But sometimes documents may reflect multiple semantic meanings, which represents multi-label learning. It is the method of associating a set of predefined classes to an unseen object depending on its properties. Different methods to do multi-label classification are divided into two groups, namely data transformation and algorithm adaptation. This paper focuses on the evaluation of eight algorithms of multi-label learning based on nine performance metrics using eight multi-label datasets, and evaluation is performed based on the results of experimentation. For all the multi-label classifiers used for experimentation, decision tree is used as a base classifier whenever required. Performance of different classifiers varies according to the size, label cardinality, and domain of the dataset.

Keywords Machine learning · Multi-label classification · Data transformation Algorithm adaptation · Decision tree · Label cardinality

1 Introduction

One commonly used task in mining is *classification*. If a set of known instances, called train set, is used to train the model, then it is referred as *supervised learning*. Once the training and testing of the model are complete, it is useful for classification

V. S. Tidake (✉)
Department of Computer Engineering, MCERC, Savitribai Phule Pune University, Nashik, India
e-mail: vaishalitidake@yahoo.co.in

S. S. Sane
Department of Computer Engineering, KKWIEER, Savitribai Phule Pune University, Nashik, India
e-mail: sssane@kkwagh.edu.in

S. Bhalla et al. (eds.), *Intelligent Computing and Information and Communication*,
Advances in Intelligent Systems and Computing 673,
https://doi.org/10.1007/978-981-10-7245-1_13

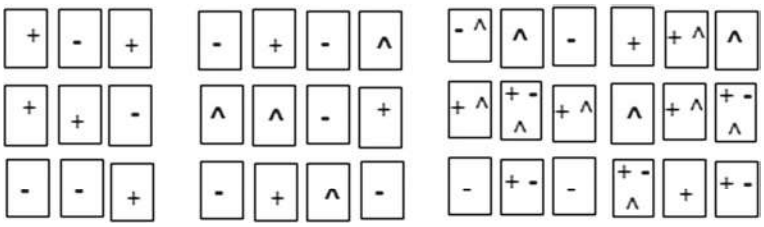
of unseen instances. Several distinct domains [1–8] like TC use supervised learning. Sometimes, a document may reflect multiple semantic meanings. Hence, unlike traditional classification, it may be associated with one or more than one class labels, which represents multi-label learning. It is the method of associating a set of predefined classes to an unseen document depending on its contents. Association of each input example with single-class label is termed as *SL (single-label) classification* or just classification. Depending on the total count of class labels involved, SL classification is either referred as a *binary single-label (BSL) classification* when the label space has only two class labels or *multi-class single-label (MSL) classification* if the label space includes more than two class labels. For example, a news document represented as a square in Fig. 1 may be related to either education (+) or health (−) category representing BSL (Fig. 1a) or one of education, health, and economy (^) categories representing MSL (Fig. 1b) [2, 6]. A news saying that "Yoga and meditation are crucial for the stress management of students" is related to education as well as health categories (+ −) representing *MLC (multi-label)* classification (Fig. 1c). Already many tools and algorithms are available to handle SL classification problems. Use of MLC in the recent past has been done for TC, prediction of gene function, tag recommendation, discovery of drug, [2–6], etc. So in the area of machine learning, it has gained the position of an upcoming research field.

This paper deals with a comparative study of MLC. Sections 2, 3, and 4 describe the metrics used for evaluation, two approaches used for MLC, about the experiments and results, respectively. Section 5 gives the concluding remarks.

2 Multi-label Classification (MLC)

A. Definition

Like SL, MLC uses supervised approach for learning. It is the task which relates an unseen instance considering its features to a set of predefined labels. Let C represents a set of disjoint labels. Let an instance be described by a vector f_j of



(a) BSL classification, (b) MSL classification, (c) ML classification

Fig. 1 Classification

features and belongs to a set y_j of labels. Let Q denotes a training set (x_j, y_j), and then obtain a function $f(x)$ for mapping each f_j vector to a set y_j of labels, where $y_j \subseteq C$ and $j = 1, 2\ldots |Q|$.

B. **Metrics used to measure performance**

Let PL_i and AL_i denote set of predicted labels by a classifier and a set of actual labels for training instance x_i. Let T and C denote a test set and a set of disjoint labels, respectively. Let f denotes a classifier. ML learning uses metrics following metrics.

Hamming loss Most commonly used which is used to measure the number of times an instance and its associated label is not correctly classified. Expected value of hamming loss metric is small [2].

$$HL(f) = \frac{1}{|T|}\sum_{i=1}^{T}\frac{|B(PLi\ \Theta\ ALi)|}{|C|}, \tag{1}$$

where $B(.) = 0$ if AL_i and PL_i are same for all labels of instance i, else $B(.) = 1$. Here, Θ is used for symmetric difference.

Ranking loss This metric measures performance of ranking task which generates all labels in the order of relevance. It is used to measure the number of times an irrelevant label has been ranked above the relevant labels. Expected value of ranking loss metric is small [6].

$$RL(f) = \frac{1}{|T|}\sum_{i=1}^{|T|}\frac{1}{|ALi|\left|\overline{ALi}\right|}\left|\{(y1, y2)|\mu(y1, xi) \geq \mu(y2, xi)\}\right|, \tag{2}$$

where $y1 \in ALi$ and $y2 \in \overline{ALi}$. Assume $\mu(q, r)$ denotes relevance of label q for an instance r and smaller value denotes more relevance.

One-error It counts the number of times a label generated by the classifier at the top rank does not appear in the correct labels associated with an input instance. The smaller the one-error, the better it is [6, 7]. Here, $B(.) = 1$ if (.) is true, else $B(.) = 0$.

$$OE(f) = \frac{1}{|T|}\sum_{i=1}^{T} B\left(\left(\arg\min_{y\in C}\mu(y, xi)\right) \notin ALi\right) \tag{3}$$

Coverage It measures how much down the list of labels generated by the classifier should be traversed to include all the labels relevant to an example assuming top-most labels appear at the start of the list. The less the value, the better is the result [6, 7].

$$CG(f) = \frac{1}{|T|}\sum_{i=1}^{|T|}\max_{y\in ALi}\mu(y,xi) - 1 \tag{4}$$

Average precision It computes an average proportion of relevant labels which are ranked above a particular relevant label. The bigger the value, The better is the result [6, 7].

$$AP(f) = \frac{1}{|T|}\sum_{i=1}^{T}\frac{1}{|ALi|}\sum\nolimits_{y\in ALi}\frac{|\{z\in ALi|\mu(z,xi)\le\mu(y,xi)\}|}{\mu(y,xi)} \tag{5}$$

Subset Accuracy It is an average over all the instances which checks whether predicted label set of an instance is same as its actual label set [3, 5, 9].

$$SA(f) = \frac{1}{|T|}\sum_{i=1}^{T}B(PLi = ALi), \tag{6}$$

where $B(.) = 1$ if AL_i and PL_i are same for all labels of instance i, else $B(.) = 0$.

Example-Based Recall, Precision, and F-Measure [2, 6, 7]:

$$\begin{aligned} ExRc(f) &= \frac{1}{|T|}\sum_{i=1}^{|T|}\frac{|PLi\cap ALi|}{|ALi|},\\ ExPr(f) &= \frac{1}{|T|}\sum_{i=1}^{T}\frac{|PLi\cap ALi|}{|PLi|},\\ ExF1(f) &= \frac{1}{|T|}\sum_{i=1}^{|T|}\frac{2|PLi\cap ALi|}{|ALi|+|PLi|}\cdots \end{aligned} \tag{7}$$

3 Various Methods

In the literature, various methods to perform multi-label learning have been developed and reported. Two broad categories used to perform MLC are the data transformation and the algorithm adaptation [4]. The data transformation approach involves transformation of an input instance into data which suits for many single-label traditional classifiers, whereas the algorithm adaptation approach involves transformation of SL classifier algorithm which suits multi-label data [2, 3].

So far various SL algorithms are developed by researchers. The data transformation approach utilizes these existing SL algorithms. The approach transforms data representation from multi-label to single-label which is acceptable by existing

SL classification algorithms. In different words, the data transformation operates on the fundamental concept of "*fit data to an algorithm*" [2]. Transformation does not change the algorithm, hence is said to be "*independent of an algorithm*" [3]. Algorithms like BR, LP, CC, ECC, RAkEL, and HOMER use this approach. The adaptation approach modifies the existing SL algorithms for managing multi-label data appropriately. In different words, the algorithm adaptation operates on the fundamental concept of "*fit an algorithm to data*" [2]. Since an algorithm, not data, is updated, this approach is said to be "*dependent on an algorithm*" [3]. Algorithms like BRkNN, MLkNN, ML-C4.5, and BP-MLL use this approach. Let C represents a set of labels.

Binary Relevance (BR)
It is the most widely used method for data transformation in which a multi-label problem is converted into $|C|$ binary SL classification problems. Each of the binary classifiers contributes its vote separately to do classification [4, 5]. BR has one disadvantage of not considering the association between labels (if any) as it treats every label individually [2, 6].

Label Powerset (LP)
Overcoming the drawback of BR for treating every label individually is removed in LP. It considers each different group of labels as a separate class and treats the entire problem as a multi-class single-label (MSL) problem [7].

Multi-label data is treated as multi-class data. For example, multi-label data having $|C|$ labels forms $2^{|C|}$ classes with different label combinations. Thus, LP considers multiple labels simultaneously and overcomes the drawback of BR [8]. However, the number of groups of classes formed increases with $|C|$. It results in distribution of the original data into different groups of classes. This distribution may result in scenario similar to class imbalance where few classes may belong to more number of instances, whereas some classes may belong to less number of instances. The situation may affect classifier accuracy. Also, higher value of C causes time complexity of LP to become worst.

Random k-Label sets (RAkEL) It is actually an ensemble of multiple LP classifiers having different combinations of all labels referred as label sets [1]. These k-size label sets help to remove the class imbalance drawback in LP. For multi-label data with $|C|$ labels, N label sets each of size k are formed randomly and separate LP classifier models are designed for them. Average of votes obtained from N models for each label is used for classification of an unseen instance. If it is more than a threshold, then prediction of that label is *P;* otherwise, it is *A* to represent the presence or absence of that label. However, classifier accuracy depends on the label sets which are selected randomly. Also, choosing N and k values may also affect the classifier performance.

Classifier Chain (CC) A weakness of BR not considering association between labels is removed in CC [7, 8]. Like BR, it transforms ML problem into $|C|$ SL problems and for each label C_j, a separate binary classifier B_j is designed. But the input for each classifier B_j is different. Each classifier B_j takes as input all feature

vectors $f_{1\ldots D}$ of all instances and predictions of all earlier classifiers also. In general, output O_{ij} of each classifier represents prediction of classifier B_i for $Class_i$ for instance j. O_{ij} takes values either P or A for class C_i of instance j. Accordingly, output of all classifiers is obtained. Thus, label information is passed from classifier B_i to B_j, and so on. Such organization takes into account associations among labels and thus overcomes the weakness of BR described earlier. But an important concern in CC is that sequence of considering labels may result in different classifier accuracies [8] affecting its performance to a great extent and guessing the best possible order is difficult.

Ensemble of Classifier Chains (ECC) Instead of depending on single chain of labels, ECC takes benefit of using multiple different order chains as well as ensemble. It obtains votes from a group of classifiers each using different chains and different set of instances, which improves the accuracy of prediction [8].

Calibrated Label Ranking (CLR) It is a modification of Ranking by Pairwise Comparison (RPC) [2, 6]. It augments the label set with a virtual label L_v. Then, it constructs $C(C-1)/2$ binary classifiers as in RPC. Each classifier B_{ij} outputs P for an instance if it contains label C_i and A if it contains label C_j, and does not consider instances having both or none of these labels in the pair (C_i, C_j) [5]. CLR also constructs C binary classifiers to represent relationship between each label C_i and a virtual label L_v. While classifying an unseen instance, votes are obtained from all these constructed classifiers to generate ranking of all labels having relevant and irrelevant labels separated by a virtual label.

Multi-Label k-Nearest Neighbors (MLkNN) It updates traditional kNN algorithm to process multi-label data. For classification of an unseen instance, it finds k-nearest neighbors. Then, statistical data like count of nearest neighbors for a training instance x associated with particular label and not associated with particular label is obtained for each label using computed k-nearest neighbors. Next, a rule based on Bayes theorem is applied to labels of an unseen instance [2, 10]. Further, it computes label information from obtained nearest neighbors with the help of posterior and prior probabilities. The MLkNN exhibits a limitation of not considering label relationship by processing each label separately.

Hierarchy Of Multi-labEl leaRners (HOMER) Each individual classifier works on smaller size distinct label set as compared to the original one, where each label set contains related labels together. Hierarchical distribution of these labels is an important feature of this classifier [2].

4 Experimentation and Results

A. Multi-label Datasets

MEKA is a WEKA-based project. An open-source library Mulan uses Java to perform multi-label data mining. Data sets from various domains are made available in MEKA, MULAN, and LibSVM [11–14]. Table 1 briefs some multi-label datasets used for experimentation along with their information [11]. Table 1 shows label cardinality of all datasets which denotes the average number of labels per example [2].

B. Parameter initialization

For BR, LP, CLR, CC, and ECC, C4.5 decision tree algorithm [15, 12] is used as base SL classifier. For HOMER and RAkEL, LP with C4.5 is used as a base classifier. HOMER is run with three clusters and random method. RAkEL [1] runs with ix models, 3 as size of subset and 0.5 as threshold. MLkNN is executed with 10 neighbors and 1 as smoothing factor. Cross-validation is used for evaluation with tenfolds [9, 11].

C. Results and Discussion

Experiments are carried out using Intel(R) Core(TM) i5-6200U CPU @2.30 GHz having 8 GB RAM and Windows10 and Java for programming with libraries from Mulan 1.5 [11] and WEKA 3.8.1 [12]. Figures 2, 3, 4, 5, 6, 7, 8, 9, and 10 show results obtained from execution of eight classifiers on eight datasets to measure nine performance metrics. Information in brackets shows criteria expected for that metric value. Legend for all charts is same as that shown in Fig. 10.

LP classifier produced memory error for mediamill dataset when running on the above-mentioned hardware. An attempt is done to compare results obtained with other work reported in the literature [7, 11, 16]. The variation in the results may be due to different base classifiers used or different parameter settings or different default parameters in different tools used for experimentation. For multimedia

Table 1 Multi-label datasets

Domain	Dataset	#attributes	#labels	#instances	Label cardinality	Label density
Biology	Yeast	103	14	2417	4.237	0.30
Biology	Genbase	1186	27	662	1.252	0.05
Text	Medical	1449	45	978	1.245	0.03
Text	Enron	1001	53	1702	3.378	0.06
Multimedia	Scene	294	6	2407	1.073	0.18
Multimedia	Corel5 k	499	374	5000	3.522	0.01
Multimedia	Emotions	72	6	593	1.868	0.31
Multimedia	Mediamill	120	101	43,907	4.375	0.04

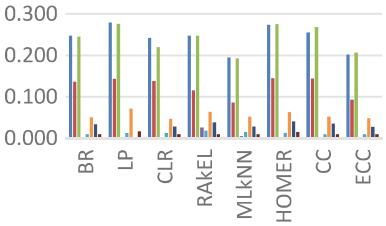


Fig. 2 Hamming loss (small)

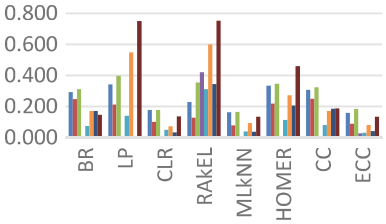


Fig. 3 Ranking loss (small)

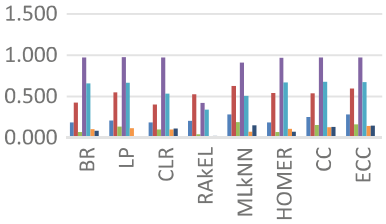


Fig. 4 Subset accuracy (large)

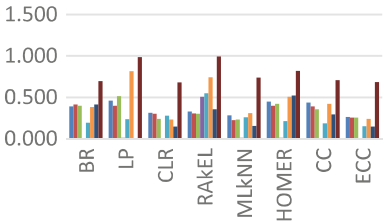


Fig. 5 One-error (small)

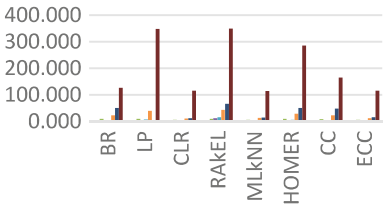


Fig. 6 Coverage (small)

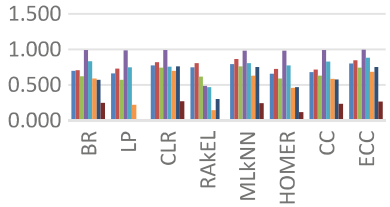


Fig. 7 Average precision (large)

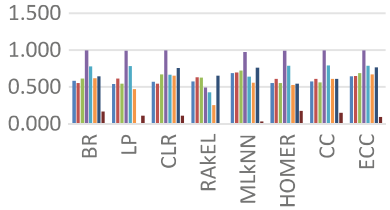


Fig. 8 Example-based precision (large)

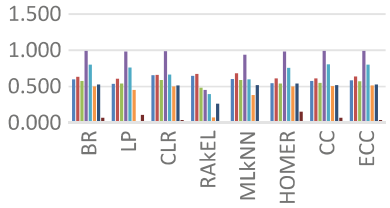


Fig. 9 Example-based recall (large)

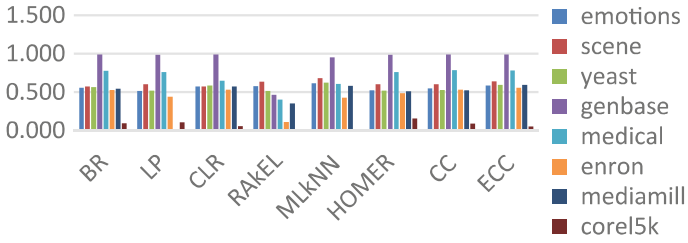


Fig. 10 Comparison of various classifiers for example-based F-measure (large)

domain, MLkNN achieved best results in terms of hamming loss followed by subset accuracy. Especially, mediamill and corel5 k datasets of multimedia domain having more label cardinality achieved better results on hamming loss and subset accuracy

among all. Emotions dataset in multimedia domain showed good performance for all MLC methods using C4.5 [15] except LP and HOMER for all example-based metrics, namely hamming loss, example-based recall, precision, and F-measure as compared to ranking-based metrics, namely coverage, one-error, and average precision [7]. LP, RAkEL, and HOMER performed poor in all domains for all metrics. The reason can be that the decision tree classifier may not be able to utilize the label relationship to the required extent when used with LP. Drawback of LP showing poor performance if different classes are associated with different numbers of examples should not be ignored. Also, subset size and number of models used with RAkEL [1] may be too small for not considering label correlation properly. Already, LP with decision tree has shown poor results and same is taken as base classifiers for HOMER and RAkEL, which may be the reason behind their poor performance. ECC performed much better than CC, LP, CLR, RAkEL, and HOMER for all domains next to MLkNN for ranking loss, hamming loss, coverage, and one-error, but poor for the remaining metrics. It can be due to the power of ensemble which has been already proved to be better than single-label classifier in the literature [8]. For biology domain, BR, CC, and ECC showed best performance on all example-based measures followed by CLR for genbase dataset only. Yeast dataset has shown poor performance than genbase dataset. We can hypothesize that it may be due to higher label cardinality for yeast dataset as compared to genbase dataset. Similar may be the case for the text domain. Medical dataset in text domain has achieved better metric values especially in BR, CLR, CC, and ECC for the same reason as compared to enron dataset. MLkNN and CLR showed less misclassification of instance label pairs by giving smaller hamming loss than [7]. The cause may be different in number of neighbors and the base classifier selected in both classifiers, respectively.

5 Conclusion

There are two ways to design algorithms for ML classification. Data transformation methods transform data having multiple labels into single-label aiding traditional single-label methods. The adaptation approach updates the existing algorithms of learning for processing multi-label data. For all the multi-label classifiers used in this work, decision tree is used as a base SL classifier wherever necessary. Mediamill and corel5 k of multimedia domain having more label cardinality achieved good results on hamming loss and subset accuracy. For biology domain, BR, CC, and ECC showed best performance on all example-based measures on both datasets followed by CLR for genbase only. Medical in text domain has achieved better metric values especially in BR, CLR, CC, and ECC as compared to enron. ECC also showed effectiveness next to MLkNN but better than other methods, thus giving their votes to ensemble and adaptation, respectively. It will be interesting to see the effect of other base classifiers on different ML classifiers.

References

1. 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.
2. 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.
3. 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.
4. G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," Int. J. Data Warehousing Mining, vol. 3, no. 3, pp. 1–13, 2007.
5. 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.
6. 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.
7. 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.
8. 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.
9. Nasierding, Gulisong, and Abbas Z. Kouzani. "Comparative evaluation of multi-label classification methods." In Fuzzy Systems and Knowledge Discovery (FSKD), 2012 9th International Conference on, pp. 679–683. IEEE, 2012.
10. M. -L. Zhang and Z. -H. Zhou, "ML-KNN: A lazy learning approach to multi-label learning," *Pattern Recognit.*, vol. 40, no. 7, pp. 2038–2048, 2007.
11. G. Tsoumakas, E. Spyromitros-Xioufis, J. Vilcek, and I. Vlahavas, "MULAN: A Java library for multi-label learning," *J. Mach. Learn. Res.*, vol. 12, pp. 2411–2414, Jul. 2011.
12. M. Hall *et al.,* "The WEKA data mining software: An update", *SIGKDD Explor.*, vol. 11, no. 1, pp. 10–18, 2009.
13. C. -C. Chang and C. -J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, Article 27, 2011 [Online]. Available: http://www.csie.ntu.edu.tw/∼cjlin/libsvm.
14. Tidake, Vaishali S., and Shirish S. Sane. "Multi-label Learning with MEKA", CSI Communications (2016).
15. Ross Quinlan (1993). C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, San Mateo, CA.
16. Cerri, Ricardo, Renato RO da Silva, and André CPLF de Carvalho. "Comparing methods for multilabel classification of proteins using machine learning techniques." In Brazilian Symposium on Bioinformatics, pp. 109–120. Springer Berlin Heidelberg, 2009.