Multi-Label Classification via Manipulating Labels

Huaping Guo, Ming Fan
School of Information Engineering, ZhengZhou University, P. R. China
hpguo.gm@gmail.com, mfan@zzu.edu.cn

***Abstract*—Unlike traditional classification problem, multi-label learning task is to predict a label set with unknown size for an example. While the exponential number of possible label sets challenges the task of multi-label learning. Many approaches by manipulating labels have been proposed. In this paper, we propose a new method via manipulating labels for multi-Label Learning: adding a virtual label to the original label set, appending the label subset selected by mutual information for each pairwise labels to the original feature set, and finally learning a binary classifier for each pairwise labels. Extensive experiments show that, compared with advanced multi-label methods, the proposed method induces models with significantly better performance..**

Keywords- multi-label, mutual information, pairwise labels

I. INTRODUCTION

Unlike traditional classification, multi-label learning associate an example with multiple labels and its task is to predict the proper labels for the unseen example.

Multi-label learning widely exists in practical problems such as each document including several topics (economics, volunteers and so on) and one image containing several objects (building, sunset, sea, trees and so on).

However, one problem existing in multi-label learning problem is efficiency. In principle, given a label set with M labels, there are 2^M - 1 label combination candidates for an unseen example, which is computational infeasible by enumerating all candidates to find the best one. Another equally important is label imbalance. Since examples of the rare label occur infrequently, many existing models can not effectively detect the examples.

Many multi-label learning methods have been proposed to tackle these problem. These methods can be roughly grouped into two categories: those based on problem transformation [1, 2] and those based on algorithm adaptation [3]. The former transforms the learning task into one or more single-label classification tasks, each of which a traditional single-label classifier learning method (e.g. k-nearest neighbor [2]) is applied to. The latter extends a specific traditional method (e.g. decision tree) to handle multi-label data directly [3].

In this paper, we contribute a new multi-label learning method called MCML (Multi-label Classification via Manipulating Labels) based on problem transformation. MCML firstly adds a virtual label to the original label set and sort the label set. Then, for each pairwise labels (c_i, c_j), the labels selected by feature selection methods (e.g. mutual information) are also treated as traditional attribute. Finally, MCML learns a binary classifier for each pairwise labels (c_i, c_j). For prediction, all the classifiers vote the corresponding labels and the final model predicts an unseen example based on the votes. The experimental results show that, compared with other state-of-the-art methods, MCML induces models with significantly better performance.

The remainder of this paper is structured as follows: after reviewing the related work in next section, section III introduces the proposed metric. Section IV presents the experimental results, and finally, section V concludes this paper.

II. RELATED WORK

A. Feature Selection and Mutual Information

Feature selection is a fundamental method in data mining to select an optimal/suboptimal feature subset and cast away irrelevant redundant features from an original feature set. Many metrics have been proposed to evaluate the correlation of two or more features (variables), in which mutual information is among the most popular ones [4].

The mutual information of two random variables is a quantity that measures the mutual dependence of two variables. Formally, the mutual information of two discrete random variables X and Y is defined as:

$$I(X,Y)=\sum_{x\in\chi}\sum_{y\in\gamma}p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \qquad (1)$$

where p(x, y) is the joint probability distribution function of the two discrete variables X and Y, and p(x) and p(y) are the marginal probability distribution functions of X and Y respectively. If I(X, Y) is large, the mutual dependence of variables X and Y is large, otherwise small.

In practice, mutual information calculated by Eq. 1 is susceptible to noisy data. Su and Zhang [5] use a threshold to affirm the reliability of dependence of two variables, defined as:

$$\rho(X,Y)=\frac{\lambda\log N}{2N}T_X^Y \qquad (2)$$

where N is the number of observed examples, λ is a constant and $T_X^Y=|\chi||\gamma|$. If I(X, Y) > ρ(X, Y), the dependence between X and Y is significant.

In this paper, we apply this metric based feature selection method as candidate to multi-label learning problem, though many metrics based method can be adopted immediately, that will be one of our future work.

Published by Atlantis Press, Paris, France.

B. Multi-label Learning

This section only focuses on learning methods based on problem transformation, since the proposed method in this paper also belongs to this type.

Algorithm 1: TRAINNING

Input: training dataset D, and the mutual information $I(c_i, c_j)$
Output: classifier H

```
1: for each class c_i ∈ {c_0, c_1,...c_K} do
2:    for each class c_j ∈ {c_{i+1}, c_{i+2},...c_K} do
3:          L_ij = {};
4:          for each class c_l ∈ {c_0, c_1,...c_K} do
5:            if I(c_l, c_i) > ρ(c_l, c_i) ^ I(c_l, c_j) > ρ(c_l, c_i) then //refer to Eq. 2
6:                    L_ij = L_ij ∪ {c_l};
7:              end if
8:          end for
9:          D' ={};
10:         for instance (x, Y) ∈ D do
11:           if y_i ≠ y_j then
12:                 D_ij = D_ij ∪ {(x, L_ij), (y_i, y_j)};
13:           end if
14:         end for
15:       training tbinary classifier H_ij for pairwise (c_i, c_j) on D_ij;
16:    end for
17: end for
18: return H;
```

Algorithm 2: CLASSIFY

Input: unseen example x, and classifier H;
Output: predicted label set Z

```
1: N = 0; //the vector whose each element N_i is the number of votes of H
          on class c_i
2: Z = 0;
3: for each classifier H_ij ∈ H do
4:          N_{H_ij} ++;
5: end for
6: for each class c_i ∈ {c_1, c_2,...c_K} do
7:          if N_i ≥ N_0 then
8:                  z_i = 1;
9:          end if
10:end for
11:return Z;
```

The most widely employed method of problem transformation is Binary Relevance (BR) [1] which learns one binary classifier H_i for each label $c \in \{c_1, c_2, ... c_n\}$, i.e., H_i: $x \rightarrow \{\neg c_i, c_i\}$, where x is an example described by feature set X. BR produces the final decision by combining the predictions of all classifiers on an unseen example. In this way, BR achieves fast learning and prediction. However it suffers from the label independence assumption, and fails to take advantage of any relations between labels.

As an extension version of BR, Classifier Chain (CC) [6] utilizes the correlation of labels by ordering label set $L = \{c_1, c_2, ...c_n\}$. Like BR, CC trains one classifier H_i for each label c_i, i.e., H_i: $x \rightarrow \{\neg c_i, c_i\}$. Unlike BR, when training the classifier for label c_i, CC treats the former labels (c_1, c_2, c_{i-1}) as features.

Another approach of utilizing the correlation of labels is called calibrated label ranking (CLR) [7,8]. Unlike CC, CLR first adds a virtual label to label set and then learning a classifier H_{ij} for each pairwise label set (c_i, c_j), $i \neq j$, i.e., H_{ij}: $x \rightarrow \{(\neg c_i, c_j), (c_i, \neg c_j)\}$. For prediction, all the classifiers vote the corresponding labels and the final result can be got according to the votes on each label.

This paper proposes a new method called MCML to utilize the correlation of labels for multi-label learning. Unlike BR, MCML trains a classifier for each pairwise labels set, as CLR does. Unlike CLR, MCML treats the labels forehead pairwise labels as features, as BL does. Besides, MCML applies feature selection technique to tackle with the features obtained from labels to increase model performance, that is different from both BR and CLR.

III. Proposed Method

Let $D = \{x_i | i = 1, 2,...n\}$ be an example set where each example x_i is associated with a indicator vector $Y_i=(y_1, y_2,... y_K)$, $y_i \in \{0, 1\}$. $y_j = 1$ if c_j is an desired label associated with x_i and 0 otherwise. Let $Z_i = \{z_1, z_2...z_K\}$ be an indicator vector in which $z_j \in \{0, 1\}$ indicates whether label c_j is predicted to be a real label associated with example x_i. Therefore, the diversity between the desired label set associated with each example and the real label set predicted on the example can be obtained according to Y_i and Z_i.

The main idea of MCML is as follows. Firstly MCML adds a virtual label c_0 to label set $C=\{c_1, c_2,... c_K\}$ and orders the label set. For simplicity, we assume that the ordered label set is $C=\{c_0, c_1,...c_K\}$. Then MCML iterative trains a classifier for each pairwise labels (c_i, c_j), $0 \leq i < j \leq K$. For current pairwise labels (c_i, c_j), it appends the label subset obtained from $\{c_1, c_2,...c_{i-1}\}$ by Eq. 1 (mutual information) and Eq. 2 to feature set, and then learns a classifier H_{ij} for the pairwise labels on dataset $D_{ij} \subseteq D$ obtained by selecting the examples each of which one and only one of the labels c_i and c_j is associated to. For prediction, each classifier H_{ij} votes an unseen example to be c_i or c_j. The label with votes number greater than the votes on label c_0 is predicted to be a real label of this unseen example.

The specific details of MCML's training procedure are shown in algorithm 1. MCML iteratively builds a classifier for each pairwise labels. For current pairwise (c_i, c_j), it first constructs new features for the pairwise using Eq. 1 and Eq. 2 (lines 4-8), then constructs data set D_{ij} for training classifier H_{ij} (Lines 10-14), and finally training H_{ij} (line 15). The details of MCML's classification procedure is shown in algorithm 2. It first statistics the votes of H on example x (lines 3-5), and then decides whether each class c_i is a label of x (lines 6-10). If the votes on c_i is not less than the votes on virtual label c_0, c_i is a label of x. Otherwise, c_i is not a label of x.

IV. Experiments

A. Evaluation Metrics

Four evaluation metrics are employed to evaluate the performance of the proposed method [1]: Hamming-loss, One-error, Coverage and Ranking-loss.

The Hamming loss is defined as:

$$Hamming-loss = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i \Delta Z_i|}{|L|}$$

where $Y_i=(y_1, y_2,... y_K)$, $y_i \in \{0, 1\}$ is an indicator vector: $y_j = 1$ if c_j is an desired label associated with x_i and 0 otherwise. $Z_i = \{z_1, z_2...z_K\}$ be an indicator vector in which $z_j \in \{0, 1\}$ indicates whether label c_j is predicted to be a real label associated with example x_i, as defined before. $\triangle$ stands for the symmetric difference of two sets, which is the set-theoretic equivalent of the exclusive disjunction (XOR operation) in Boolean logic.

One-error evaluates how many times the top-ranked label is not in the set of relevant labels of the instance

$$1 - Error = \frac{1}{m}\sum_{i=1}^{m}\delta\left(\arg\min_{c_h \in L} r_i(c_h)\right)$$

where $\delta(\lambda) = 1 if\ c_h \notin R_i, 0 otherwise$

Coverage evaluates how far we need, on average, to go down the ranked list of labels in order to cover all the relevant labels of the example.

$$Cov = \frac{1}{m}\sum_{i=1}^{m}\max_{\lambda \in Y_i} r_i(c) - 1$$

Coverage is normalized to [0, 1] by Cov/|L|.

Ranking loss expresses the number of times that irrelevant labels are ranked higher than relevant labels:

$$R\text{–}loss = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{|Y_i||\overline{Y_i}|}\left|(\lambda_a,\lambda_b): r_i(\lambda_a) > r_i(\lambda_b), (\lambda_a,\lambda_b) \in Y_i \times \overline{Y_i}\right|$$

The smaller the four metric, the better the corresponding algorithm is.

Tabel 1. The details of experimental data sets

DataSet	*Dom*	*m*	*|F|*	*|L|*	LC	LD
medical	text	978	1449	45	1.245	0.028
genbase	biology	662	1186	27	1.252	0.046
yeast	biology	2417	103	14	4.237	0.303
emotions	music	593	72	6	1.869	0.311
scene	image	2407	294	6	1.074	0.179
enron	text	1702	1001	53	3.378	0.064

B. Data Sets

Six multi-label data sets obtained from four domains (text, biology, music and image) are used for experiments to evaluate the performance of the proposed method. The details of the data sets are shown in Table 1: the name of data set (DataSet), the domain (Dom), instances size (m), the number of labels (|L|), feature numbers (|F|), label cardinality (LC) and label density (LD), where cabel cardinality of a dataset D is the average number of labels of the examples:

$$Label - Cardinality = \frac{1}{m}\sum_{i=1}^{m}|Y_i|$$

and Label density of D is the average number of labels of the examples in *D* divided by q:

$$Label - Density = \frac{1}{m}\sum_{i=1}^{m}\frac{|Y_i|}{|L|}$$

These data sets can be download from http://mulan.sourceforge.net/datasets.html. For each set, a ten-fold cross-validation is performed.

C. Experiment Setup and Results

CC [1], CLR [7, 8] and MLKNN[2] are selected as candidate comparing method to evaluate the performance of MCML, where the base classifier is J48, which is a Java implementation of C4.5 [9] from Weka [10]. In MLKNN, we set k =10 and set Euclidean distance as default distance measure. The four popular measures discussed in section IV.II are selected for evaluating the performance of multi-label learning methods, i,e. hamming loss, one-error, coverage and ranking loss. The smaller the value of the four measures, the better the performance of the corresponding algorithm. Furthermore, we set λ=5 in Eq. 2 and use |L| (the size of label set) to normalize Coverage to be a value in [0,1].

The performance results in terms of hamming loss, one-error, coverage and ranking loss are shown in Table 2, Table 3, Table 4 and Table 5 respectively, where •/◦ next to a result indicates that MCML is significantly better/worse than the respective method (column) for the respective data set (row) with pairwise t-test at 5% significance level. As shown in the Table 2, Table 3, Table 4 and Table 5, MCML significantly outperforms other advanced multi-label learning methods on all of the four metrics, since MCML employs the virtues of between CC and CRL, and applies feature selection technique to remove redundant feature to improve classifier accuracy. This results also validates the conclusion: reasonably utilizing label correlation can improve the performance of multi-label classifiers [1].

V. CONCLUSIONS

This paper contributes a new multi-label method which employs both pair-wise ranking technique and feature selection technique for multi-label classification. Empirical results show that the proposed method to construct multi-label classifiers leads to significantly better accuracy results compared to state-of-the-art methods.

As discussion in section II.I (Feature Selection and Mutual Information), many feature selection methods can be applied to the proposed method immediately. Therefore, one of our future work is to extensively study the effectiveness of feature selection on multi-label learning. Many label ranking methods which we have not discussed in this paper, and thus another our future work is to explore the relationship between feature selection and label ranking.

References

[1] G. Tsoumakas, I. Katakis and I. Vlahavas, “Mining Multi-label Data Data”, Mining and Knowledge Discovery Handbook, Springer, 2nd edition, 2010.

[2] M.L. Zhang and Z.H. Zhou, “A k-nearest neighbor based algorithm for multi-label classification”, IEEE International Conference on Granular Computing, IEEE, 2005.

[3] C. Vens, J. Struyf, L. Schietgat, S. Dzeroski and H. Blockeel, “Decision trees for hierarchical multi-label classification”, Machine Learning, 2008.

[4] P. Chanda, Y. R. Cho, A. Zhang and M. Ramanathan, “Mining of Attribute Interactions Using Information Theoretic Metrics”, IEEE International Conference on Data Mining Workshops, 2009

[5] J. Su and H. Zhang, "Full bayesian network classifiers", Proc. 23rd International Conference on Mechine Learning , 2006.

[6] J. Read, B. Pfahringer, G. Holmes and E. Frank, "Classifier chains for multi-label classification", ECML/PKDD 2009, pages 254-269, Springer, 2009.

[7] K. Brinker, J. Furnkranz, E.L. Hullermeier: "A unified model for multilabel classification and ranking", Proceedings of the 17th European Conference on Artificial Intelligence, 489-493, 2006.

[8] J. Furnkranz, E. Hullermeier, E.L. Mencia, K. Brinker, "Multilabel classification via calibrated label ranking", Machine Learning 2008.

[9] J.R. Quinlan, C4.5: programs for machine learning. Morgan Kaufmann, 1993.

[10] I.H. Witten, E. Frank, Data Mining: Practical Machine Learning Tools and Techniques. Second ed, Morgan Kaufmann, 2005.

Table 2. Performance (mean±std) of each algorithm in terms of hamming loss

Data set	MCML	CC	CLR	MLKNN
medical	0.0100±0.0009	0.0102±0.0017	0.0104±0.0009 •	0.0151±0.0020 •
genbase	0.0013±0.0008	0.0011±0.0006	0.0013±0.0008	0.0050±0.0027 •
yeast	0.1838±0.0056	0.2682±0.0071 •	0.2202±0.0091 •	0.1933±0.0123 ◦
emotions	0.2249±0.0282	0.2550±0.0181 •	0.2423±0.0286 •	0.1951±0.0243 •
scene	0.1023±0.0093	0.1444±0.0164 •	0.1383±0.0089 •	0.0862±0.0084 ◦
enron	0.0427±0.0019	0.0524±0.0024 •	0.0471±0.0019 •	0.0524±0.0020 •

•denotes that MCML outperforms compared algorithms with pairwise t-test at 5% significance level and ◦ is outperformed by compared algorithms. The notations in the following tables have the same meaning.

Table 3. Performance (mean±std) of each algorithm in terms of one error

Data set	MCML	CC	CLR	MLKNN
medical	0.1432±0.0333	0.1862±0.0404 •	0.1637±0.0267 •	0.2403±0.0465 •
genbase	0.0015±0.0045	0.0030±0.0061	0.0015±0.0045	0.0136±0.0158
yeast	0.2097±0.0312	0.3562±0.0222 •	0.2412±0.0379 •	0.2292±0.0336
emotions	0.2378±0.0538	0.4353±0.0447 •	0.3154±0.0751 •	0.2835±0.0740 •
scene	0.1645±0.0227	0.3914±0.0453 •	0.3020±0.0404 •	0.2239±0.0304 •
enron	0.1546±0.0289	0.4201±0.0289 •	0.2315±0.0436 •	0.3050±0.0280 •

Table 4. Performance (mean±std) of each algorithm in terms of ranking loss

Data set	MCML	CC	CLR	MLKNN
medical	0.0387±0.0093	0.1029±0.0238 •	0.0404±0.0090 •	0.0586±0.0153 •
genbase	0.0015±0.0045	0.0238±0.0126 ◦	0.0238±0.0126	0.0204±0.0115
yeast	0.4531±0.0108	0.6316±0.0164 •	0.4786±0.0171 •	0.4452±0.0146
emotions	0.2834±0.0337	0.4225±0.0381 •	0.3134±0.0242 •	0.2981±0.0272
scene	0.0629±0.0088	0.2251±0.0334 •	0.0985±0.0108 •	0.0791±0.0091 •
enron	0.1991±0.0179	0.4366±0.0340 •	0.2119±0.0164 •	0.2475±0.0190 •

Table 5. Performance (mean±std) of each algorithm in terms of coverage

Data set	MCML	CC	CLR	MLKNN
medical	0.0246±0.0091	0.0812±0.0235 •	0.0261±0.0091 •	0.0403±0.0099 •
genbase	0.0082±0.0057	0.0028±0.0036 ◦	0.0082±0.0057	0.0060±0.0058
yeast	0.1459±0.0084	0.3238±0.0153 •	0.1783±0.0131 •	0.1652±0.0126 •
emotions	0.1448±0.0316	0.3066±0.0367 •	0.1784±0.0282 •	0.1633±0.0320
scene	0.0593±0.0101	0.2489±0.0370 •	0.1011±0.0135 •	0.0773±0.0116 •
enron	0.0636±0.0077	0.1719±0.0161 •	0.0715±0.0078 •	0.0922±0.0088 •