Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2013, Article ID 208051, 5 pages
http://dx.doi.org/10.1155/2013/208051

Research Article

An Ensemble Method for High-Dimensional Multilabel Data

Huawen Liu,[1,2] Zhonglong Zheng,[1] Jianmin Zhao,[1] and Ronghua Ye[1]

[1] *College of Mathematics, Physics and Information Engineering, Zhejiang Normal University, Jinhua 321004, China*
[2] *NCMIS, Academy of Mathematics and Systems Science, CAS, Beijing 100190, China*

Correspondence should be addressed to Huawen Liu; hwliu@zjnu.edu.cn

Received 1 August 2013; Accepted 5 October 2013

Academic Editor: Zhiqiang Ma

Multilabel learning is now receiving an increasing attention from a variety of domains and many learning algorithms have been witnessed. Similarly, the multilabel learning may also suffer from the problems of high dimensionality, and little attention has been paid to this issue. In this paper, we propose a new ensemble learning algorithms for multilabel data. The main characteristic of our method is that it exploits the features with local discriminative capabilities for each label to serve the purpose of classification. Specifically, for each label, the discriminative capabilities of features on positive and negative data are estimated, and then the top features with the highest capabilities are obtained. Finally, a binary classifier for each label is constructed on the top features. Experimental results on the benchmark data sets show that the proposed method outperforms four popular and previously published multilabel learning algorithms.

1. Introduction

Data classification is one of the major issues in data mining and machine learning. Generally speaking, it consists of two stages, that is, building classification models and predicting labels for unknown data. Depending on the number of labels tagged on each data, the classification problems can be divided into single-label and multilabel classification [1]. In the former, the class labels are mutually exclusive and each instance is tagged with only one class label. On the contrary, each instance may be tagged with more than one class label simultaneously. The multilabel classification problems are ubiquitous in real-world applications, such as text categorization, image annotation, bioinformatics, and information retrieval [1, 2]. For example, the movie "*avatar*" may be tagged with *action*, *science fiction,* and *love* types.

Now, many multilabel classification algorithms have been witnessed. Roughly speaking, they can be grouped into two categories, that is, algorithm adaption and problem transformation [1]. The first kind of technique extends traditional single-label classifiers, such as *k*NN, C4.5, SVM, and AdaBoost, by modifying some constraint conditions to handle multilabel data. Typical examples include AdaBoost.MH [3], BR*k*NN [4], and LP*k*NN [4]. For instance, Zhang and Zhou [5] proposed ML*k*NN and applied tscene classification, while Clare and King [2] employed C4.5 to deal with multilabel data by altering the discriminative formula of information entropy.

The second technique of multilabel learning transforms multilabel data into corresponding single-label ones and then handle them one by one using the traditional methods. An intuitive approach is to treat the multilabel problem as a set of independent binary classification problems, one for each class label [6, 7]. However, they often have not considered the correlations among the class labels and may suffer from the problem of unbalanced data, especially when there are a large number of the class labels [8]. To cope with these problems, several strategies have been introduced. For example, Zhu et al. [9] explored the label correlation with maximum entropy, while Cai and Hofmann [10] captured the correlation information among the labels by virtue of a hierarchical structure.

Analogous to traditional classification, multilabel learning may also encounter the problems, such as over-fitting and the curse of dimensionality, raised from high dimensionality of data [11, 12]. To alleviate this problem, an effective solution is to perform dimension reduction or feature selection on data in advance. As a typical example, Ji et al. [13] extracted

a common subspace shared among multiple labels by using ridge regression. One common characteristic of these methods is that they make use of only one feature set to achieve the learning purpose under the context of multilabel data. However, in reality, only one feature subset can not represent the properties of different labels exactly. Therefore, it is necessary to choose different features for each label during the multilabel learning stage. A representative example of such kind is LIFT [14].

In this paper, we propose a new multilabel learning algorithm. The main characteristic of our method is that during the procedure of constructing binary classifiers different feature subsets will be exploited for each label. More specifically, given a class label, the features with high discriminative capabilities with respect to the label are chosen and then used to train a binary classifier. This means that the selected features have local properties. Note that they may have lower discriminative capabilities with respect to other class labels. Other binary classifiers can also be constructed in a similar manner. Finally, all binary classifiers are assembled into an overall one, which will be used to predict or classify the labels of unknown data.

The rest of this paper is organized as follows. We describe the details of the proposed method in Section 2. Experimental results conducted to evaluate the effectiveness of our method are presented in Section 3. Finally, conclusions and future works are given in the end.

2. Binary Classification with Feature Selection

Assume that $L = \{l_1, \ldots, l_m\}$ denotes the finite set of labels in a multilabel learning task. Let $\mathscr{D} = \{(\mathbf{x}_1, \mathbf{y}_1), (\mathbf{x}_2, \mathbf{y}_2), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$ be a multilabel data set. It consists of n independently identically distributed (iid) samples. $\mathbf{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})^T \in X$ and $\mathbf{y}_i = (y_{i1}, y_{i2}, \ldots, y_{im})^T \in Y$ are the d-dimensional feature and m-dimensional label vectors of the ith sample, respectively. y_{ij} takes a value of 1 or 0, indicating whether is the ith sample associated with the jth label or not.

2.1. Binary Classification. According to the formal description of $\mathscr{D}$, we know that the multilabel data is a general form of traditional single-label data, whereas $\mathbf{y}_i$ only involves one single label. Thus, a natural and intuitive solution for multilabel learning is to transform the multilabel data into its corresponding single-label data and then train classifiers on the generated data. There are many transformation strategies. *Copy*, *selection,* and *ignore* are three typical transformation techniques [1]. Besides, the power set of labels is also introduced in the literature, where every $\mathbf{y}_i$ is often taken as a new class label.

Before giving the principle of binary classification, let us introduce the concepts of positive and negative samples of labels.

Definition 1. Given a multilabel data set $\mathscr{D}$ with n samples associated with m labels L, for each class label $l_k \in L$, its positive samples $P(l_k)$ and negative samples $N(l_k)$ are defined as follows:

$$P(l_k) = \{\mathbf{x}_i \mid (\mathbf{x}_i, \mathbf{y}_i) \in \mathscr{D}, y_{ik} = 1\}, \quad (1)$$

$$N(l_k) = \{\mathbf{x}_i \mid (\mathbf{x}_i, \mathbf{y}_i) \in \mathscr{D}, y_{ik} = 0\}. \quad (2)$$

From this definition, we know that, given a label l_k, all examples of the original data set are positive if they are associated with the class label l_k and negatively otherwise. Moreover, $P(l_k) \cap N(l_k) = \emptyset$ and $P(l_k) \cup N(l_k) = X$.

Binary relevance (BR), also known as one-against-all method, is the most popular and most commonly used transformation method for multilabel learning in the literature [1]. It learns m different binary classifiers independently, one for each different label in L. Specifically, it transforms the original data set $\mathscr{D}$ into m data sets $\mathscr{D}_i$, $i = 1, .., m$. Each data set $\mathscr{D}_i$ consists of the positive samples $P(l_i)$ and negative samples $N(l_i)$ with respect to l_i. Based on the new data set $\mathscr{D}_i$, a binary classifier f_i for the label l_i can be built using the off-the-shelf learning methods, for example, kNN and SVM. After obtaining m binary classifiers for all labels, the prediction of BR for a new sample x is the union of the labels l_k that are positively predicted by the m classifiers; that is, $f(x) = [f_1(x), f_2(x), \ldots, f_m(x)]^T$, where $f_i(x)$ takes a value of 0 or 1, indicating x is predicted positively or negatively by the classifier f_i.

BR is a straight forward transformation method and widely used as a baseline in comparson with multilabel learning algorithms. However, the drawback of BR is that it does not take correlations among the labels into account and treats all labels independently. In addition, it also suffers from the class imbalance problem. In multilabel data, the number of positive samples $P(l_k)$ is significantly less than the number of negative samples $N(l_k)$ for some labels due to the typical sparsity of labels. To alleviate this problem, feature selection should be performed on the data sets in advance.

2.2. Feature Selection. The purpose of feature selection is to select significant features to represent data from the original space without losing information greatly. It has been extensively studied in the traditional learning. However, little work of feature selection has been done in the context of multilabel learning. Currently, there are many criteria available to measure the interestingness of features [15]. Here, we exploit the concept of density distribution of data to represent the interestingness of features.

Definition 2. Given a data set $\mathscr{D}_k$ with n samples, the density of value distribution of the ith feature is defined as

$$\rho_i^k(\mathscr{D}_k) = \frac{\sum_{u=1}^{n}\sum_{v=1}^{n} \operatorname{sim}(x_{ui}, x_{vi})}{n \cdot \min_{1 \le u,v \le n} \operatorname{sim}(x_{ui}, x_{vi})}, \quad (3)$$

where x_{ui} denotes the ith feature value of x_u and sim is a similar function between two values.

In (3), the sim function is often taken as the form of inverse Euclidean distance. If the positive samples $P(l_k)$ and negative samples $N(l_k)$ are considered in (3), we can get the positive and negative densities of features.

Definition 3. Given the positive samples $P(l_k)$ and negative samples $N(l_k)$, the positive and negative densities of the ith feature are defined as

$$\rho_{i+}^{k} = \rho_{i}^{k}\left(P\left(l_k\right)\right), \tag{4}$$

$$\rho_{i-}^{k} = \rho_{i}^{k}\left(N\left(l_k\right)\right). \tag{5}$$

The positive density ρ_{i+}^{k}, as well as the negative density ρ_{i-}^{k}, can effectively represent the specific characteristic of data. The larger the value of ρ_{i+}^{k} (or ρ_{i-}^{k}), the better discriminative capability to distinguish positive (or negative) samples from others.

Based on this principle, we adopt these two criteria to choose significant features during the learning stage. Specifically, for each feature a_i in $\mathscr{D}_k$, we calculate its positive density ρ_{i+}^{k} and negative density ρ_{i-}^{k}, respectively. Then, the positive densities of all features will be ranked in a decreasing order, and the top t features with high positive densities will be selected. Similar situation can be done for the negative densities. Finally, the features with high positive and negative densities will be used to train desirable binary classifiers.

How many features should be selected for classification is still an open problem. Here, we empirically determine the number of selected features with a concept called m_k-minimum density, which is defined in the following.

Definition 4. Let $\mathscr{D}_k$ be the data set, and let $P(l_k)$ and $N(l_k)$ be the positive and negative samples of the ith feature, respectively. The m_k-minimum density of $\mathscr{D}_k$ with respect to l_k is

$$m_k = r \cdot \min\left(\left|P\left(l_k\right), N\left(l_k\right)\right|\right), \tag{6}$$

where $r \in [0, 1]$ and $|\cdot|$ is the set cardinality.

The m_k-minimum density can effectively measure the information amount that one feature has. If the density is larger than the m_k-minimum density, the corresponding feature has enough information to represent the characteristics of data. As a result, the feature will be chosen during the stage of feature selection. In other words, after calculating ρ_{i+}^{k} and ρ_{i-}^{k}, we retain the features with ρ_{i+}^{k} or ρ_{i-}^{k} larger than m_k and discard the others. Note that the parameter r in Definition 4 is to control the number of selected features. The larger value of r is, the more features would be chosen. In our empirical experiments, the classifier achieved good performance when r was set to 0.1.

2.3. The Proposed Method. Based on the analysis above, we propose a new multilabel learning algorithm. The framework of our algorithm is shown as Algorithm 1. The proposed method works in a straightforward way and can be easily understood. It consists of two major stages, that is, learning and prediction stages. In the training stage, a new data set will be generated for each class label by obtaining its positive and negative samples. Subsequently, we estimate the interestingness of features in the data set, so as to retain significant features for classification. Finally, a binary classifier is constructed with a baseline learning method. Given a new sample, its class labels can be predicted by testing it with all binary classifiers.

3. Empirical Study

To validate the performance of our proposed method, we made a comparson ofEMCFS with three popular multilabel learning algorithms. They are ML-kNN, LIFT, and Rank-SVM, standing for different kinds of learning types. In addition, we took a linear support vector machine as the baseline binary classification algorithm and assigned 0.1 to the parameter r. Other parameters were set as their default values suggested by authors. For example, the number of the nearest neighbors in ML-kNN was 10 and the distance measure is the Euclidean distance [5]. In the Rank-SVM classifier, the degree of polynomial kernels was 8 and the cost parameter c was assigned as one [16].

3.1. Data Sets. To validate the effectiveness of our method roundly, our experiments have been conducted on four data sets, including *emotions*, *medical*, *corel16k* (sample 1), and *delicious*. They are often used to verify the performance of multilabel classifiers in the literature and are available at http://mulan.sourceforge.net/datasets.html. Table 1 summarizes their general information, where the *cardinality* and *density* columns refer to the average number of class labels of the samples and its fraction by the number of labels. These multilabel data sets vary from the quantities of labels and differ greatly in the sizes of samples and features [17].

3.2. Experimental Results. There are lots of evaluation criteria available to evaluate the performance of multilabel classifiers. In this work, average precision, hamming loss, one error, and coverage have been adopted to assess the effectiveness of the proposed method. Their detailed descriptions can be found in several literatures such as those in [1, 18].

Table 2 reports the average precision of the multilabel classifiers on the data sets. In in this table, each row denotes an observation on the data sets. The best result comparable with others in the same row is highlighted in boldface, where the larger the value, the better the performance. From the table, one may observe that the proposed method, EMCFS, works quite well and is comparable to others in most cases with the average precision. For example, on the *delicious* data set (the last row in in the table), the precision of EMCFS is 29.5%, which is the best one among the others.

Apart from the average precision, we also compared EMCFS to the others from the perspective of the hamming loss, one error, and coverage. Tables 3, 4, and 5 present the averaged performance of the learning algorithms in terms of these three criteria, respectively, where the smaller the value, the better the performance. The best results are also highlighted in boldface.

According to algorithms the results in these tables, we know that similar to the average precision, EMCFS is also superior to other regarding the aspects of hamming loss, one error and coverage. Although EMCFS achieved slightly poor

Input: $\mathscr{D} = \{(\mathbf{x}_1, \mathbf{y}_1), \ldots, (\mathbf{x}_n, \mathbf{y}_n)\}$: The training multilabel data set;
r: The parameter of m_k;
$U = \mathbf{x}_1, \ldots, \mathbf{x}_s$: The test data set without labels;
Output: $\mathbf{Y} = \{\mathbf{y}_1, \ldots, \mathbf{y}_s\}$: The label set of U;
Training stage
For each label l_k in L **do**
Obtain $P(l_k)$ and $N(l_k)$ of l_k from $\mathscr{D}$ according to Def.(1)(2);
Calculate m_k with $P(l_k)$ and $N(l_k)$ according to Def. (6);
For each feature a_i in $\mathscr{D}_k = P(l_k) \cup N(l_k)$ **do**
Calculate ρ_{i+}^k and ρ_{i-}^k according to Def.(4)(5);
Select the features whose ρ_{i+}^k or ρ_{i-}^k is larger than m_k;
Train a binary classifier f_k on $\mathscr{D}_k$ with the selected features;
Endfor
Prediction stage
For each sample o_i in U **do**
$\mathbf{y}_i = \emptyset$;
For each classifier f_k **do**
$\mathbf{y}_i = \mathbf{y}_i \cup \{f_k(o_i)\}$, where f_k returns 0 or 1;
Endfor

Algorithm 1: Ensemble multilabel classifier using feature selection (EMCFS).

Table 1: The brief description information of data sets in experiments.

Data set	Domain	Sample	Feature	Label	Cardinality	Density
Emotions	Music	593	72	6	1.87	0.31
Medical	Text	978	1449	45	1.25	0.03
Corel16k	Image	13766	500	153	2.86	0.02
Delicious	Web	16105	500	983	19.02	0.02

Table 2: A comparison of average precision of four classifiers on data sets (%).

Data set	EMCFS	ML-kNN	LIFT	Rank-SVM
Emotions	**80.5**	76.3	78.9	77.6
Medical	**89.9**	78.2	86.9	87.2
Corel16k	**32.3**	27.6	31.5	29.7
Delicious	**29.5**	25.7	28.2	22.4

Table 3: A comparison of hamming loss of four classifiers on data sets.

Data set	EMCFS	ML-kNN	LIFT	Rank-SVM
Emotions	**0.189**	0.214	0.206	0.218
Medical	**0.009**	0.016	0.012	0.014
Corel16k	**0.017**	0.928	0.017	0.019
Delicious	**0.019**	0.998	0.021	0.025

Table 4: A comparison of one error of four classifiers on data sets.

Data set	EMCFS	ML-kNN	LIFT	Rank-SVM
Emotions	**0.253**	0.285	0.264	0.293
Medical	**0.140**	0.267	0.180	0.194
Corel16k	**0.671**	0.852	0.674	0.783
Delicious	**0.401**	0.482	0.433	0.665

Table 5: A comparison of coverage of four classifiers on data sets.

Data set	EMCFS	ML-kNN	LIFT	Rank-SVM
Emotions	**2.148**	2.561	2.302	2.335
Medical	**1.256**	2.504	1.403	1.852
Corel16k	140.428	151.853	139.592	**136.527**
Delicious	**662.746**	674.613	667.513	671.652

coverage on the *corel16k* data set, the performance is 140.428, which is slightly worse than the best one. However, it is not the worst in comparson with ML-kNN.

4. Conclusions

In this paper, we propose a new ensemble multilabel learning method. The central idea of our method is that, for each label, it exploits different features to build learning models. The advantage is that the classifiers are constructed on the features with strong local discriminative capabilities. Generally, the proposed method consists of three steps. Firstly, for each label, a new data set is generated by identifying the positive and negative samples. Then, the interestingnes's of features will be estimated and the features with high density will be retained to train a learning model. Finally, all binary classifiers built with the selected features will be integrated into an overall one. Experimental results on four multilabel

data sets show that the proposed method can potentially improve performance and outperform other competing and popular methods.

Acknowledgments

The authors are grateful to the anonymous referees for their valuable comments and suggestions. This work is partially supported by the National NSF of China (61100119, 61170108, 61170109, 61272130, and 61272468), the NSF of Zhejiang province (Y1110483), Postdoctoral Science Foundation of China (2013M530072), and the Open Project Program of the National Laboratory of Pattern Recognition (NLPR) (201204214).

References

[1] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Mining multi-label data," in *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds., pp. 667–686, 2010.

[2] A. Clare and R. King, "Knowledge discovery in multi-label phenotype data," in *Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, pp. 42–53, 2001.

[3] R. E. Schapire and Y. Singer, "BoosTexter: a boosting-based system for text categorization," *Machine Learning*, vol. 39, no. 2, pp. 135–168, 2000.

[4] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1079–1089, 2011.

[5] M.-L. Zhang and Z.-H. Zhou, "ML-KNN: a lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, pp. 2038–2048, 2007.

[6] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1-2, pp. 1–39, 2010.

[7] M. L. Zhang and Z. H. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, no. 99.

[8] G. M. Weiss and F. Provost, "Learning when training data are costly: the effect of class distribution on tree induction," *Journal of Artificial Intelligence Research*, vol. 19, pp. 315–354, 2003.

[9] S. Zhu, X. Ji, W. Xu, and Y. Gong, "Multi-labelled classification using maximum entropy method," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 274–281, 2005.

[10] L. Cai and T. Hofmann, "Hierarchical document categorization with Support Vector Machines," in *Proceedings of the Thirteenth ACM Conference on Information and Knowledge Management (CIKM '04)*, pp. 78–87, November 2004.

[11] H. Liu, X. Wu, and S. Zhang, "A new supervised feature selection method for pattern classification," *Computational Intelligence*, 2013.

[12] B. Bonev, F. Escolano, D. Giorgi, and S. Biasotti, "Information-theoretic selection of high-dimensional spectral features for structural recognition," *Computer Vision and Image Understanding*, vol. 117, no. 3, pp. 214–228, 2013.

[13] S. Ji, L. Tang, S. Yu, and J. Ye, "A shared-subspace learning framework for multi-label classification," *ACM Transactions on Knowledge Discovery from Data*, vol. 4, no. 2, article 8, 2010.

[14] M. L. Zhang, "LIFT: multi-label learning with label-specific features," in *Proceedings of the 22nd international joint conference on Artificial Intelligence (IJCAI '11)*, pp. 1609–1614, 2011.

[15] X. Li, J. Wang, J. Zhou, and M. Yin, "A perturb biogeography based optimization with mutation for global numerical optimization," *Applied Mathematics and Computation*, vol. 218, no. 2, pp. 598–609, 2011.

[16] A. Elisseeff and J. Weston, "A kernel method for multi-labelled classification," in *Proceedings of the Neural Information Processing Systems*, pp. 681–687, 2002.

[17] X. Li and M. Yin, "An opposition-based differential evolution algorithm for permutation flow shop scheduling based on diversity measure," *Advances in Engineering Software*, vol. 55, pp. 10–31, 2013.

[18] X. Li and M. Yin, "Application of differential evolution algorithm on self-potential data," *PloS ONE*, vol. 7, no. 12, Article ID e51199, 2012.