Multi-Label Learning by Instance Differentiation

Min-Ling Zhang and **Zhi-Hua Zhou**
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210093, China
{zhangml, zhouzh}@lamda.nju.edu.cn

Abstract

Multi-label learning deals with ambiguous examples each may belong to several concept classes simultaneously. In this learning framework, the inherent ambiguity of each example is *explicitly* expressed in the output space by being associated with *multiple* class labels. While on the other hand, its ambiguity is only *implicitly* encoded in the input space by being represented by only a *single* instance. Based on this recognition, we hypothesize that if the inherent ambiguity can be explicitly expressed in the input space appropriately, the problem of multi-label learning can be solved more effectively. We justify this hypothesis by proposing a novel multi-label learning approach named INSDIF. The core of INSDIF is *instance differentiation* that transforms an example into a bag of instances each of which reflects the example's relationship with one of the possible classes. In this way, INSDIF directly addresses the inherent ambiguity of each example in the input space. A two-level classification strategy is employed to learn from the transformed examples. Applications to automatic web page categorization, natural scene classification and gene functional analysis show that our approach outperforms several well-established multi-label learning algorithms.

Introduction

Multi-label learning problems widely exist in real-world applications. For instance, in text categorization, each document may belong to several predefined topics, such as *government* and *health* (McCallum 1999; Schapire & Singer 2000); in functional genomics, each gene may be associated with a set of functional classes, such as *metabolism*, *transcription* and *protein synthesis* (Elisseeff & Weston 2002); in scene classification, each scene image may belong to several semantic classes, such as *beach* and *urban* (Boutell *et al.* 2004). When solving these multi-label learning problems, each example (a real-world object) in the training set is represented by a single instance associated with a set of labels, and the task is to output a label set whose size is unknown *a priori* for the unseen example.

It is obvious that multi-label learning deals with ambiguous objects, i.e., objects which have different semantic meanings simultaneously if they are viewed from different aspects. Previous approaches to multi-label learning mainly include decomposing the task into multiple independent binary classification problems each for a class (Joachims 1998; Yang 1999), considering the ranking quality among labels (Schapire & Singer 2000; Crammer & Singer 2002; Elisseeff & Weston 2002; Zhang & Zhou 2006) and exploring the class correlation (McCallum 1999; Ueda & Saito 2003; Ghamrawi & McCallum 2005; Yu, Yu, & Tresp 2005; Zhu *et al.* 2005; Liu, Jin, & Yang 2006). A common characteristic of these methods is that they only deal with the ambiguity of objects in the *output space* (label space), i.e., they only exploit the information conveyed by the multiple labels of each example.

In this paper, we propose a new solution to multi-label learning by considering the ambiguity of objects in the *input space* (instance space). We assume that the reason for an object belonging to several semantic classes simultaneously is essentially due to the diverse information embodied in the object. For instance, a document being labeled with topics *government* and *health* must contain sentences or sections describing information about government and health respectively; a natural scene image being labeled with types *beach* and *urban* must contain sub-images characterizing information about beach and urban respectively. However, in multi-label learning, this diverse information is only encoded in a single instance representing the object. Therefore, we expect that if the ambiguous object can be properly represented by *a bag of instances* instead of a single instance, where each instance in the bag explicitly reflects some information contained in the object from a certain aspect, a more effective solution to the task of multi-label learning may be yielded.

Based on the above recognition, a two-stage algorithm, INSDIF, which is based on *instance differentiation*, is proposed. In the first stage, INSDIF transforms each example into a bag of instances in order to explicitly express the ambiguity of the example in the input space. Briefly, for each possible class c_l, a prototype vector $\mathbf{v}_l$ is calculated by averaging all the instances belonging to c_l. After that, each example $\mathbf{t}$ is re-represented by a bag of instances each of which equals $\mathbf{t} - \mathbf{v}_l$, i.e. the difference between the example and the prototype vector of class c_l. In the second stage, a two-level classification strategy is utilized to learn from the transformed data set. Applications to three real-world tasks

show that INSDIF achieves better performance than several well-established multi-label learning algorithms.

We start the rest of this paper by briefly reviewing related works on multi-label learning. Then, we propose the INSDIF approach and report on the applications to automatic web page categorization, natural scene classification and gene functional analysis, which is followed by conclusion.

Multi-Label Learning

Let $\mathcal{X} = \mathbb{R}^d$ denote the input space and $\mathcal{Y} = \{1, 2, \ldots, Q\}$ denote the finite set of possible labels, respectively. Given a multi-label training set $S = \{(\mathbf{x}_1, Y_1), (\mathbf{x}_2, Y_2), ..., (\mathbf{x}_N, Y_N)\}$, where $\mathbf{x}_i \in \mathcal{X}$ is a single instance and $Y_i \subseteq \mathcal{Y}$ is the label set associated with $\mathbf{x}_i$, the goal of multi-label learning is to learn a function $h : \mathcal{X} \to 2^{\mathcal{Y}}$ from S which predicts a set of labels for an unseen example.

Traditional two-class and multi-class problems can both be cast into multi-label problems by restricting that each instance has only one label. On the other hand, the generality of multi-label problems inevitably makes it more difficult to address. An intuitive approach to solving multi-label problems is to decompose it into multiple independent binary classification problems (one per class). However, this kind of method does not consider the correlation between the different labels of each instance and the expressive power of such a system could be weak (Elisseeff & Weston 2002; McCallum 1999; Schapire & Singer 2000).

Majority studies on multi-label learning focus on text categorization. Most of the multi-label text categorization algorithms were derived from traditional learning techniques such as probabilistic generative models (McCallum 1999; Ueda & Saito 2003), boosting methods (Schapire & Singer 2000), decision trees (Comité, Gilleron, & Tommasi 2003), maximum entropy methods (Ghamrawi & McCallum 2005; Zhu *et al.* 2005), bayes decision rules (Gao *et al.* 2004), neural networks (Crammer & Singer 2002; Zhang & Zhou 2006), *k*-nearest neighbor methods (Zhang & Zhou 2007), and maximal margin methods (Godbole & Sarawagi 2004; Kazawa *et al.* 2005). Several studies aim to improve the performance of text categorization systems by exploiting additional information given by the hierarchical structure of classes (Cai & Hofmann 2004; Rousu *et al.* 2005) or unlabeled data (Liu, Jin, & Yang 2006).

In addition to text categorization, multi-label learning has also manifested its effectiveness in many other real-world applications, such as bioinformatics (Clare & King 2001; Elisseeff & Weston 2002; Brinker, Fürnkranz, & Hüllermeier 2006; Barutcuoglu, Schapire, & Troyanskaya 2006; Brinker & Hüllermeier 2007), scene classification (Boutell *et al.* 2004), and association rule mining (Thabtah, Cowling, & Peng 2004; Rak, Kurgan, & Reformat 2005). It is worth noting that although most works on multi-label learning assume that an instance is associated with multiple valid labels, there is also a study that assumes that only one of the labels associated with an instance is correct (Jin & Ghahramani 2003).

The INSDIF Approach

As stated before, in multi-label learning, although the inherent ambiguity of the object is explicitly expressed in the output space by having multiple labels, it is vaguely implied in the input space by having only a single instance. In this section, we propose to explicitly express the ambiguity in the input space by automatically transforming the single instance representation into a *bag* representation. After that, the induced learning problem is solved by a two-level classification method.

In the first stage, INSDIF computes a prototype vector $\mathbf{v}_l$ for each class c_l by averaging all the instances in the training set which belong to c_l:

$$\mathbf{v}_l = \left(\sum\nolimits_{\mathbf{x}_i \in U_l} \mathbf{x}_i\right) / |U_l|, \text{ where}$$
$$U_l = \{\mathbf{x}_i | \{\mathbf{x}_i, Y_i\} \in S,\ l \in Y_i\},\ l \in \mathcal{Y} \quad (1)$$

Here $\mathbf{v}_l$ can be approximately regarded as a profile-style vector describing common characteristics of class c_l. Actually, this kind of prototype vectors have already shown their effectiveness in solving text categorization problems. Specifically, the ROCCHIO method (Ittner, Lewis, & Ahn 1995; Sebastiani 2002) forms a prototype vector for each class by averaging all the documents (represented by weight vectors) of this class, and then classifies the test document by calculating the dot-products between the weight vector representing the document and each of the prototype vectors. In this paper, this kind of prototype vectors are also utilized to facilitate bag generation. After acquiring prototype vectors, each example $\mathbf{x}_i$ is re-represented by a bag of instances B_i, where each instance in B_i is just the difference between $\mathbf{x}_i$ and one prototype vector:

$$B_i = \{\mathbf{x}_i - \mathbf{v}_l | l \in \mathcal{Y}\} \quad (2)$$

In this way, each example is transformed into a bag whose size equals to the number of possible classes.

In fact, such a process attempts to exploit the spatial distribution since $\mathbf{x}_i - \mathbf{v}_l$ in Eq. 2 is a kind of distance between $\mathbf{x}_i$ and $\mathbf{v}_l$. The transformation can also be realized in other ways. For example, other than referring to the prototype vector of each class, maybe we can go with the following way: For each possible class c_l, identify the *k*-nearest neighbors of $\mathbf{x}_i$ among training instances with class c_l. Then, the mean vector of these neighbors can be regarded as an instance in the bag. Note that the transformation of a single instance to a bag of instances can be realized as a general pre-processing method that can be plugged into many machine learning systems.

In the second stage, INSDIF learns from the transformed training set $S^{new} = \{(B_1, Y_1), (B_2, Y_2), \ldots, (B_N, Y_N)\}$. Actually, this kind of learning problem falls into the recently proposed learning framework of *multi-instance multi-label learning* (Zhou & Zhang 2007), where each example is associated with not only multiple instances but also multiple class labels. So, the two-level learning process used in INSDIF can also be regarded as a new multi-instance multi-label learning method.

The idea of pre-processing examples before learning has also been employed by the *constraint classification* (CC)

method (Har-Peled, Roth, & Zimak 2003). In this method, label information of each example is encoded by a set of constraints each of which specifies the relative order between a pair of classes for this example. There are two main differences between the pre-processing schemes used by INSDIF and CC. Firstly, for INSDIF, each example is transformed into a fixed number (i.e. Q) of instances each is with dimensionality d; While for CC, the number of instances transformed from each example is decided by the number of the constraints, and each instance is with dimensionality Qd. Secondly, for INSDIF, the induced learning problem after transformation is in fact solved by a multi-instance multi-label learner; While for CC, the induced learning problem after transformation can be solved by any *binary* classifier.

Figure 1 shows the two-level classification structure employed by INSDIF. Input to the structure is a bag B consisting of n instances $\{\mathbf{b}_1, \mathbf{b}_2, \ldots, \mathbf{b}_n\}$, where each instance $\mathbf{b}_k$ is a d-dimensional feature vector $[\mathbf{b}_{k1}, \mathbf{b}_{k2}, \ldots, \mathbf{b}_{kd}]^{\mathrm{T}}$. Outputs of the structure consist of Q real values $\{y_1, y_2, \ldots, y_Q\}$, where each output y_l corresponds to a label $l \in \mathcal{Y}$. The first level is composed of M bags $\{C_1, C_2, \ldots, C_M\}$, where each bag C_j is the medoid of group G_j. Here $\{G_1, G_2, \ldots, G_M\}$ partition the transformed training set into disjoint *groups of bags* with $\bigcup_{j=1}^{M} G_j = \{B_1, B_2, \ldots, B_N\}$ and $G_i \bigcap_{i \neq j} G_j = \emptyset$. The second level weights $\mathbf{W} = [w_{jl}]_{M \times Q}$ connect each medoid C_j in the first level to each output y_l.

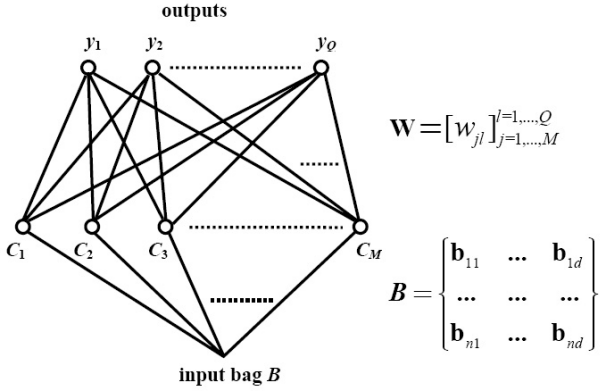

Figure 1: Two-level classification structure used by INSDIF

Firstly, by regarding each bag as an atomic object, the popular k-medoids algorithm is adapted to cluster the transformed training set into M disjoint groups of bags. In this paper, we employ Hausdorff distance (Edgar 1995) to measure distance between bags. Formally, given two bags of instances $A = \{\mathbf{a}_1, \ldots, \mathbf{a}_{n_1}\}$ and $B = \{\mathbf{b}_1, \ldots, \mathbf{b}_{n_2}\}$, the Hausdorff distance between A and B is defined as:

$$\mathrm{H}(A, B) = \max\{\max_{\mathbf{a} \in A} \min_{\mathbf{b} \in B} \|\mathbf{a} - \mathbf{b}\|, \max_{\mathbf{b} \in B} \min_{\mathbf{a} \in A} \|\mathbf{b} - \mathbf{a}\|\}$$

where $\|\mathbf{a} - \mathbf{b}\|$ measures the distance between instances **a** and **b**, which takes the form of Euclidean distance here. Note that categorical data can be processed by adopting appropriate distance metric, such as the Value Difference Metric (VDM) (Stanfill & Waltz 1986).

Y = INSDIF(S, M, **z**)

Inputs:

S : the multi-label training set $\{(\mathbf{x}_1, Y_1), \ldots, (\mathbf{x}_N, Y_N)\}$

M : the number of medoids in the first level

z : the test example ($\mathbf{z} \in \mathcal{X}$)

Outputs:

Y : the predicted label set for **z** ($Y \subseteq \mathcal{Y}$)

Process:

1 Compute prototype vectors $\mathbf{v}_l$ ($l \in \mathcal{Y}$) using Eq. 1;
2 Form the new training set S^{new} by transforming each $\mathbf{x}_i$ into a bag of instances B_i using Eq. 2;
3 Cluster $\{B_1, B_2, \ldots, B_N\}$ into M partitions using k-medoids algorithm combined with Hausdorff distance;
4 Determine medoid C_j of each partition using Eq. 3;
5 Compute second layer weights $\mathbf{W}$ by solving Eq. 5 using singular value decomposition;
6 Transform **z** into a bag of instances Z using Eq. 2;
7 $Y = \{l | y_l(Z) = \sum_{j=1}^{M} w_{jl} \phi_j(Z) > 0,\ l \in \mathcal{Y}\}$.

Figure 2: Pseudo-code of INSDIF

After the clustering process, the transformed training set is divided into M partitions whose medoids C_j ($1 \leq j \leq M$) are determined as:

$$C_j = \arg\min_{A \in G_j} \sum\nolimits_{B \in G_j} \mathrm{H}(A, B) \tag{3}$$

Since clustering could help find the underlying structure of a data set, the medoid of each group may encode some distributional information of different bags. With the help of these medoids, each bag B can be converted into an M-dimensional feature vector $[\phi_1(B), \phi_2(B), ..., \phi_M(B)]^{\mathrm{T}}$ with $\phi_j(B) = \mathrm{H}(B, C_j)$. The second level weights $\mathbf{W} = [w_{jl}]_{M \times Q}$ are optimized by minimizing the following sum-of-squares error function:

$$E = \frac{1}{2} \sum\nolimits_{i=1}^{N} \sum\nolimits_{l=1}^{Q} \{y_l(B_i) - d_l^i\}^2 \tag{4}$$

where $y_l(B_i) = \sum_{j=1}^{M} w_{jl} \phi_j(B_i)$ is the actual output of the structure on B_i on the l-th class, and d_l^i is the desired output of B_i on the l-th class which takes the value of +1 if $l \in Y_i$ and -1 otherwise. Differentiating the objective function in Eq. 4 with respect to w_{jl} and setting the derivative to zero gives the normal equations for the least-squares problem as follows:

$$(\mathbf{\Phi}^{\mathrm{T}} \mathbf{\Phi}) \mathbf{W} = \mathbf{\Phi}^{\mathrm{T}} \mathbf{T} \tag{5}$$

Here $\mathbf{\Phi} = [\phi_{ij}]_{N \times M}$ is with elements $\phi_{ij} = \phi_j(B_i)$ and $\mathbf{T} = [t_{il}]_{N \times Q}$ is with elements $t_{il} = d_l^i$. In this paper, the second layer weights $\mathbf{W}$ are computed by solving Eq. 5 using singular value decomposition (Press *et al.* 1992).

In summary, Figure 2 gives the complete description of our two-stage approach to multi-label learning. In the first stage (steps 1 to 2), INSDIF aims to explicitly express the ambiguity in the input space, where each example is transformed into a bag of instances by querying the class prototype vectors. In the second stage (steps 3 to 5), a two-level

Table 1: Experimental results (mean±std.) of the compared algorithms on the web page data sets. For each evaluation criterion, "↓" indicates "the smaller the better" while "↑" indicates "the bigger the better".

Evaluation Criterion	Algorithm					
	INSDIF	BOOSTEXTER	ADTBOOST.MH	RANK-SVM	CNMF	BSVM
Hamming Loss ↓	**0.039±0.013**	0.046±0.016	0.043±0.013	0.043±0.014	N/A	0.042±0.015
One-error ↓	0.381±0.118	0.446±0.139	0.461±0.137	0.440±0.143	0.509±0.142	**0.375±0.119**
Coverage ↓	4.545±1.285	4.213±1.313	**4.083±1.191**	7.508±2.396	6.717±1.588	6.919±1.767
Ranking Loss ↓	**0.102±0.037**	0.103±0.040	N/A	0.193±0.065	0.171±0.058	0.168±0.047
Average Precision ↑	**0.686±0.091**	0.638±0.108	0.632±0.105	0.605±0.117	0.561±0.114	0.660±0.093

classification strategy is used to learn from the transformed data. Finally, the test example is firstly transformed into the bag representation (step 6) and then fed to the learned classification structure for prediction (step 7).

Applications

We compare INSDIF with several state-of-the-art multi-label learning algorithms on three real-world applications. The compared algorithms include BOOSTEXTER (Schapire & Singer 2000), ADTBOOST.MH (Comité, Gilleron, & Tommasi 2003), RANK-SVM (Elisseeff & Weston 2002) and a transductive style algorithm CNMF (Liu, Jin, & Yang 2006). Moreover, BSVM, which works by decomposing the multi-label learning problem into a set of binary classification problems, is also evaluated.

For INSDIF, the parameter M as shown in Figure 2 is set to be 20% of the size of training set. [1] For BOOSTEXTER[2] and ADTBOOST.MH[3], the number of boosting rounds is set to be 500 and 50 respectively as in this paper, the performance of these two algorithms do not significantly change after the specified boosting rounds. For RANK-SVM and CNMF, the best parameters reported in (Elisseeff & Weston 2002) and (Liu, Jin, & Yang 2006) are used. For BSVM, linear kernel SVM^{light} [4] with default parameters are used as the base binary classifiers.

Automatic Web Page Categorization

The first multi-label task studied in this paper is the specific text categorization problem of WWW page categorization, which has been studied in (Ueda & Saito 2003; Kazawa *et al.* 2005; Zhang & Zhou 2007). Web pages were collected from the "yahoo.com" domain and then divided into 11 data sets based on Yahoo's top-level categories. After that, each page is classified into a number of Yahoo's second-level subcategories. Each data set contains 2,000 training documents and 3,000 test documents, where a large portion of them (about $20\% \sim 45\%$) are multi-labeled over the 11 data sets. Detailed descriptions of these 11 data sets can be found in (Zhang & Zhou 2007).

As shown in Table 1, the performance of the compared algorithms are evaluated according to five multi-label evaluation metrics, whose details can be found in (Schapire & Singer 2000). The best result on each evaluation criterion is highlighted in bold face.[5]

Table 1 shows that INSDIF performs quite well on almost all the evaluation criteria. Pairwise t-tests at 0.05 significance level reveal that INSDIF is only inferior to BSVM in terms of *one-error* and inferior to BOOSTEXTER and ADTBOOST.MH in terms of *coverage*. On the other hand, INSDIF is comparable to BOOSTEXTER and is superior to the rest algorithms in terms of *ranking loss*. More impressively, INSDIF outperforms all the other algorithms in terms of *hamming loss* and *average precision*.

Natural Scene Classification

The second multi-label task studied in this paper is natural scene classification. The data set consists of 2,000 natural scene images belonging to the classes *desert*, *mountains*, *sea*, *sunset*, and *trees*. Over 22% images belong to multiple classes simultaneously and each image is associated with 1.24 class labels on average. Detailed description of the number of images associated with different label sets can be found in (Zhang & Zhou 2007). Each image is represented by a feature vector using the same method as in (Boutell *et al.* 2004). Concretely, each color image is firstly converted to the CIE Luv space, which is a more perceptually uniform color space such that the perceived color differences correspond closely to Euclidean distances in this color space. After that, the image is divided into 49 blocks using a 7×7 grid, where in each block the first and second moments (mean and variance) of each band are computed, corresponding to a low-resolution image and to computationally inexpensive texture features respectively. Finally, each image is transformed into a $49 \times 3 \times 2 = 294$-dimensional feature vector.

Ten-fold cross-validation is performed on this data set. Table 2 reports the experimental results of the compared algorithms with the best result on each evaluation criterion highlighted in bold face. Pairwise t-tests at 0.05 significance level reveal that, in terms of all evaluation criteria, INSDIF significantly outperforms BOOSTEXTER and both of which

[1] In preliminary experiments, several percentage values have been tested ranging from 20% to 80% with an interval of 10%. The results show that these values do not significantly affect the performance of our approach.

[2] http://www.cs.princeton.edu/~schapire/boostexter.html.

[3] http://www.grappa.univ-lille3.fr/grappa/index.php3?info=logiciels.

[4] http://svmlight.joachims.org

[5] Note that *hamming loss* is not available for CNMF while *ranking loss* is not provided in the outputs of the ADTBOOST.MH implementation.

Table 2: Experimental results (mean±std.) of the compared algorithms on the natural scene image data set. For each evaluation criterion, "↓" indicates "the smaller the better" while "↑" indicates "the bigger the better".

Evaluation Criterion	Algorithm					
	INSDIF	BOOSTEXTER	ADTBOOST.MH	RANK-SVM	CNMF	BSVM
Hamming Loss ↓	**0.152±0.016**	0.179±0.015	0.193±0.014	0.253±0.055	N/A	0.202±0.015
One-error ↓	**0.259±0.030**	0.311±0.041	0.375±0.049	0.491±0.135	0.635±0.049	0.388±0.038
Coverage ↓	**0.834±0.091**	0.939±0.092	1.102±0.111	1.382±0.381	1.741±0.137	1.066±0.093
Ranking Loss ↓	**0.140±0.018**	0.168±0.020	N/A	0.278±0.096	0.370±0.032	0.196±0.022
Average Precision ↑	**0.830±0.019**	0.798±0.024	0.755±0.027	0.682±0.092	0.585±0.030	0.753±0.025

Table 3: Experimental results (mean±std.) of the compared algorithms on the Yeast data set. For each evaluation criterion, "↓" indicates "the smaller the better" while "↑" indicates "the bigger the better".

Evaluation Criterion	Algorithm					
	INSDIF	BOOSTEXTER	ADTBOOST.MH	RANK-SVM	CNMF	BSVM
Hamming Loss ↓	**0.189±0.010**	0.220±0.011	0.207±0.010	0.207±0.013	N/A	0.199±0.009
One-error ↓	**0.214±0.030**	0.278±0.034	0.244±0.035	0.243±0.039	0.354±0.184	0.227±0.032
Coverage ↓	**6.288±0.240**	6.550±0.243	6.390±0.203	7.090±0.503	7.930±1.089	7.220±0.338
Ranking Loss ↓	**0.163±0.017**	0.186±0.015	N/A	0.195±0.021	0.268±0.062	0.201±0.019
Average Precision ↑	**0.774±0.019**	0.737±0.022	0.744±0.025	0.749±0.026	0.668±0.093	0.749±0.021

are far superior to ADTBOOST.MH, RANK-SVM, CNMF and BSVM. It is also worth noting that CNMF performs quite poorly compared to other algorithms. The reason may be that the key assumption of CNMF, i.e. two examples with high similarity in the input space tend to have large overlap in the output space, does not hold on this image data set due to the big gap between low-level image features and high-level image semantics.

Yeast Gene Functional Analysis

The third multi-label task studied in this paper is to predict the gene functional classes of the Yeast *Saccharomyces cerevisiae*, which is one of the best studied organisms. Specifically, the Yeast data set investigated in (Elisseeff & Weston 2002) is used. Each gene is described by the concatenation of micro-array expression data and phylogenetic profile and is associated with a set of functional classes whose maximum size can be potentially more than 190. Actually, the whole set of functional classes is structured into hierarchies up to 4 levels deep. In this paper, as what has been done in (Elisseeff & Weston 2002), only functional classes in the top hierarchy are considered. The resulting multi-label data set contains 2,417 genes each represented by a 103-dimensional feature vector. There are 14 possible class labels and the average number of labels for each gene is 4.24 ± 1.57.

Ten-fold cross-validation is conducted on this data set. As shown in Table 3, pairwise t-tests at 0.05 significance level disclose that INSDIF performs fairly well in terms of all evaluation criteria, where on all these metrics INSDIF significantly outperforms all the other algorithms. Similarly as in nature scene classification, CNMF doesn't perform well as the basic assumption under this method may not hold on this gene data set.

Conclusion

In multi-label learning, the ambiguity of examples is explicitly expressed in the output space by associating an example with multiple labels. In this paper, we propose a new solution to multi-label learning, which attempts to explicitly express the ambiguity of examples in the input space such that the relationship between input and output ambiguities can be exploited. In the first stage, our approach transforms each example into a bag of instances, where each instance in the bag corresponds to the difference between this example and the prototype vector of a class. In the second stage, a two-level classification strategy is employed to learn from the transformed data set. Applications to three real-world multi-label tasks show that our approach achieves significant better results than several well-established multi-label learning algorithms. Investigating other ways to exploit the relationship between the input ambiguity and output ambiguity is an interesting issue for future work.

Acknowledgments

We want to thank the anonymous reviewers for their helpful comments. This work was supported by NSFC (60635030, 60473046) and the National Science Fund for Distinguished Young Scholars of China (60325207).

References

Barutcuoglu, Z.; Schapire, R. E.; and Troyanskaya, O. G. 2006. Hierarchical multi-label prediction of gene function. *Bioinformatics* 22(7):830–836.

Boutell, M. R.; Luo, J.; Shen, X.; and Brown, C. M. 2004. Learning multi-label scene classification. *Pattern Recognition* 37(9):1757–1771.

Brinker, K., and Hüllermeier, E. 2007. Case-based multilabel ranking. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 702–707.

Brinker, K.; Fürnkranz, J.; and Hüllermeier, E. 2006. A unified model for multilabel classification and ranking. In *Proceedings of the 17th European Conference on Artificial Intelligence*, 489–493.

Cai, L., and Hofmann, T. 2004. Hierarchical document categorization with support vector machines. In *Proceedings of the 13th ACM International Conference on Information and Knowledge Management*, 78–87.

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

Comité, F. D.; Gilleron, R.; and Tommasi, M. 2003. Learning multi-label alternating decision tree from texts and data. In *Proceedings of the 3rd International Conference on Machine Learning and Data Mining in Pattern Recognition*, 35–49.

Crammer, K., and Singer, Y. 2002. A new family of online algorithms for category ranking. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 151–158.

Edgar, G. A. 1995. *Measure, Topology, and Fractal Geometry, 3rd print*. Berlin: Springer-Verlag.

Elisseeff, A., and Weston, J. 2002. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems 14*. 681–687.

Gao, S.; Wu, W.; Lee, C.-H.; and Chua, T.-S. 2004. A MFoM learning approach to robust multiclass multi-label text categorization. In *Proceedings of the 21st International Conference on Machine Learning*, 329–336.

Ghamrawi, N., and McCallum, A. 2005. Collective multi-label classification. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, 195–200.

Godbole, S., and Sarawagi, S. 2004. Discriminative methods for multi-labeled classification. In *Proceedings of the 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, 22–30.

Har-Peled, S.; Roth, D.; and Zimak, D. 2003. Constraint classification for multiclass classification and ranking. In *Advances in Neural Information Processing Systems 15*. 785–792.

Ittner, D. J.; Lewis, D. D.; and Ahn, D. D. 1995. Text categorization of low quality images. In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval*, 301–315.

Jin, R., and Ghahramani, Z. 2003. Learning with multiple labels. In *Advances in Neural Information Processing Systems 15*. 897–904.

Joachims, T. 1998. Text categorization with support vector machines: learning with many relevant features. In *Proceedings of the 10th European Conference on Machine Learning*, 137–142.

Kazawa, H.; Izumitani, T.; Taira, H.; and Maeda, E. 2005. Maximal margin labeling for multi-topic text categorization. In *Advances in Neural Information Processing Systems 17*. 649–656.

Liu, Y.; Jin, R.; and Yang, L. 2006. Semi-supervised multi-label learning by constrained non-negative matrix factorization. In *Proceedings of the 21st National Conference on Artificial Intelligence*, 421–426.

McCallum, A. 1999. Multi-label text classification with a mixture model trained by EM. In *Working Notes of the AAAI'99 Workshop on Text Learning*.

Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; and Flannery, B. P. 1992. *Numerical Recipes in C: the Art of Scientific Computing*. New York: Cambridge University Press.

Rak, R.; Kurgan, L.; and Reformat, M. 2005. Multi-label associative classification of medical documents from medline. In *Proceedings of the 4th International Conference on Machine Learning and Applications*, 177–186.

Rousu, J.; Saunders, C.; Szedmak, S.; and Shawe-Taylor, J. 2005. Learning hierarchical multi-category text classfcation models. In *Proceedings of the 22nd International Conference on Machine Learning*, 774–751.

Schapire, R. E., and Singer, Y. 2000. Boostexter: a boosting-based system for text categorization. *Machine Learning* 39(2-3):135–168.

Sebastiani, F. 2002. Machine learning in automated text categorization. *ACM Computing Surveys* 34(1):1–47.

Stanfill, C., and Waltz, D. 1986. Toward memory-based reasoning. *Communications of the ACM* 29:1213–1228.

Thabtah, F. A.; Cowling, P. I.; and Peng, Y. 2004. MMAC: a new multi-class, multi-label associative classification approach. In *Proceedings of the 4th IEEE International Conference on Data Mining*, 217–224.

Ueda, N., and Saito, K. 2003. Parametric mixture models for multi-labeled text. In *Advances in Neural Information Processing Systems 15*. 721–728.

Yang, Y. 1999. An evaluation of statistical approaches to text categorization. *Information Retrieval* 1(1-2):69–90.

Yu, K.; Yu, S.; and Tresp, V. 2005. Multi-label informed latent semantic indexing. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 258–265.

Zhang, M.-L., and Zhou, Z.-H. 2006. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge and Data Engineering* 18(10):1338–1351.

Zhang, M.-L., and Zhou, Z.-H. 2007. ML-kNN: a lazy learning approach to multi-label learning. *Pattern Recognition* 40(7):2038–2048.

Zhou, Z.-H., and Zhang, M.-L. 2007. Multi-instance multi-label learning with application to scene classification. In *Advances in Neural Information Processing Systems 19*.

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