Dual Set Multi-Label Learning*

Chong Liu,[1,2] **Peng Zhao,**[1,2] **Sheng-Jun Huang,**[3] **Yuan Jiang,**[1,2] **Zhi-Hua Zhou**[1,2]
[1] National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210023, China
[2] Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210023, China
[3] College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 211106, China
{liuc, zhaop, jiangy, zhouzh}@lamda.nju.edu.cn, huangsj@nuaa.edu.cn

Abstract

In this paper, we propose a new learning framework named dual set multi-label learning, where there are two sets of labels, and an object has one and only one positive label in each set. Compared to general multi-label learning, the exclusive relationship among labels within the same set, and the pairwise inter-set label relationship are much more explicit and more likely to be fully exploited. To handle such kind of problems, a novel boosting style algorithm with model-reuse and distribution adjusting mechanisms is proposed to make the two label sets help each other. In addition, theoretical analyses are presented to show the superiority of learning from dual label sets to learning directly from all labels. To empirically evaluate the performance of our approach, we conduct experiments on two manually collected real-world datasets along with an adapted dataset. Experimental results validate the effectiveness of our approach for dual set multi-label learning.

Introduction

In many real-world tasks, an object can be naturally associated with multiple labels. Multi-label learning is proposed and studied to address such kind of problems. Given the training set, multi-label learning aims to learn a predictor which is able to classify multiple labels at the same time. For an unseen instance, the predicator is used to answer the relevant/irrelevant question on each label.

In this paper, we study a new setting where there are two sets of labels, and an object has one and only one positive label from each set. In other words, each instance always has two labels, and the labels from each of the two sets are exclusive. For example, as shown in Figure 1, the same Chinese character pronounced as 'Zhi' can be written by different people in different calligraphic fonts. Given a character from a calligraphy work, we may want to know who wrote it and which font it belongs to. Here the candidates *calligrapher* and *font* correspond to two sets of labels. In fact, this kind of problems are very common in real applications. For instance, a car can be labeled with two labels: *brand* and *type*. Given a car image, we may want to know which company produced it and which type it belongs to. Also, a movie can be annotated according to its *company* and *genre*, where companies and genres form the dual set of multiple labels. It is also noteworthy that in addition to the exclusive relationship among the labels within the same set, the inter-set relationship is usually available. For example, it is well-known that the famous Chinese calligrapher Xizhi Wang is good at running font; Porsche is good at producing sports cars; Pixar focuses on animated movies.

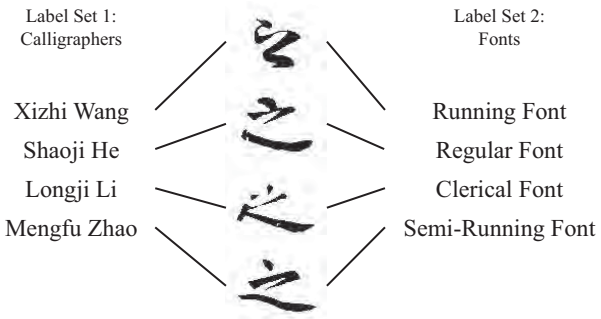


Figure 1: A real-world example of dual set multi-label learning. It shows calligraphy works of the same Chinese character 'Zhi', where calligraphers and fonts are two label sets.

Obviously, in these cases, the relationship among labels becomes much more clear than in general multi-label learning. Directly employing traditional multi-label learning algorithms to solve such problems will lead to significant disadvantages. On one hand, all labels will be equally treated, which implies that algorithm needs to decide the relevance for every label, resulting a high computational cost; on the other hand, the algorithm neglects the exclusive relationship within the label sets.

To address these issues, in this paper, we propose a novel learning framework, *dual set multi-label learning* (DSML). We also propose an effective algorithm to solve this problem based on the boosting framework. Specifically, two sample distributions are maintained for dual label sets, one for each set. And we make two base classifiers reused by each other to utilize the information embedded in the other label set. Moreover, the sample distributions are jointly adjusted such that the mistakes on one model will be made up by the other model. In this way, the proposed algorithm is expected to

*This research was supported by the NSFC (61673201), 973 Program (2014CB340501), and JiangsuSF (BK20150754).

exploit both the intra-set and inter-set label relationship simultaneously.

On all datasets collected or adapted for dual set multi-label learning, the experimental results validate the superiority of our proposed approach DSML to other compared approaches. Some diagnostic experiments are done to show the effectiveness of model-reuse and distribution adjusting mechanisms.

The main contributions of this paper include: (1) A novel machine learning framework named dual set multi-label learning is proposed; (2) An approach for DSML is proposed which outperforms other compared methods; (3) Margin analysis and generalization bound are presented to show the superiority of learning with dual label set to multi-label learning; (4) Three real-world datasets from different tasks are manually collected or adapted for DSML.

In the following, we will briefly review related works, then the dual set multi-label learning problem is formulated and some approaches are proposed to address it. Next, theoretical and experimental analyses are provided and the paper is concluded in the end.

Related Work

During the past decade, significant amount of algorithms have been proposed to deal with multi-label learning tasks (Zhang and Zhou 2014). The most straightforward way is to decompose the original problem into a series of binary classification problems, one for each label (Boutell et al. 2004); however, this solution neglects the label relationship. Previous research results (McCallum 1999) show that label relationship is very helpful and should not be neglected. Thus, some approaches (Rousu et al. 2005; Cesa-Bianchi, Gentile, and Zaniboni 2006; Hariharan et al. 2010) rely on external knowledge resources such as label hierarchies to exploit label relationship. However, external label relationship is often unavailable in practice, therefore, further approaches (Ghamrawi and McCallum 2005; Tsoumakas et al. 2009) try to exploit label relationship based on label co-occurrence. Nevertheless, directly generating label relationship from training data and then applying it to model construction may increase the risk of overfitting.

Some other approaches exploit label relationship in different ways. The basic idea of classifier chains is to transform the multi-label learning problem into a chain of binary classification problems (Read et al. 2011). Later, (Liu and Tsang 2015) and (Liu, Tsang, and Müller 2017) studied how to determine the appropriate label order for it. Instead of assuming that label correlations are shared by all instances, (Huang and Zhou 2012) exploits label correlations locally. In (Rauber et al. 2014), recursive dependent binary relevance model is proposed, where the prediction label of an instance is obtained in an iterative process.

For some algorithm adaptation methods, classical algorithms are adapted to fit multi-label learning. The basic idea behind RankSVM (Elisseeff and Weston 2002) is to fit kernel learning to multi-label data. k-nearest neighbor techniques are adapted as ML-KNN algorithm in (Zhang and Zhou 2007). BP and RBF neural networks are modified to fit multi-label learning in (Zhang and Zhou 2006) and (Zhang 2009), respectively. (Wang et al. 2016) applied linear label embedding followed by recurrent neural networks to address multi-label image classification problems. In (Yeh et al. 2017), a deep neural network based model is proposed to learn deep latent spaces for multi-label classification. Recently, (Liu and Tsang 2017) proposes a sparse coding tree framework for multi-label problems.

Boosting refers to a family of ensemble methods that are able to convert weak learners to strong learners (Zhou 2012), among which AdaBoost (Freund and Schapire 1995) is well studied and proved to be efficient in many tasks. Two boosting approaches for multi-label learning, AdaBoost.MH and AdaBoost.MR, were proposed in (Schapire and Singer 1999).

The DSML Formulation and Approaches

Problem Formulation

Let $\mathcal{X} = \mathbb{R}^d$ denote the d-dimensional input space and $\mathcal{Y} = \{\mathcal{Y}^j | j \in \{a, b\}\}$ be the label space where $\mathcal{Y}^a = \{1, \cdots, L_1\}$ denotes the label space of the first label set with L_1 possible labels and $\mathcal{Y}^b = \{1, \cdots, L_2\}$ denotes the label space of the second label set with L_2 possible labels.

Let $\mathcal{D} = \{(\boldsymbol{x}_i, y_i^a, y_i^b) | 1 \leq i \leq m\}$ denote the training set, where $\boldsymbol{x}_i$ is the feature vector for the i-th instance and $y_i^a \in \mathcal{Y}^a$, $y_i^b \in \mathcal{Y}^b$ are two labels from the two label sets respectively.

The dual set multi-label learning problem is defined as:

Definition 1. *(Dual Set Multi-Label Learning) Given the training set $\mathcal{D}$, the task is to learn a mapping function from the input space to the output space,*

$$h : \mathcal{X} \to \mathcal{Y}^a \times \mathcal{Y}^b.$$

For an unseen instance $\boldsymbol{x} \in \mathcal{X}$, the mapping function $h(\cdot)$ predicts $h(\boldsymbol{x}) \subseteq \mathcal{Y}^a \times \mathcal{Y}^b$ as the dual labels for $\boldsymbol{x}$.

Benchmark Approaches

In this part, we briefly introduce three benchmark approaches to deal with dual set multi-label learning. These approaches are based on some observations or adapted from traditional multi-label learning. All of them are problem transformation methods. Due to the page limits, more detailed information will be provided in a longer version.

Independent Decomposition is the most straightforward way to deal with dual set multi-label learning tasks. Similar to binary relevance in traditional multi-label learning, it decomposes the original dual set multi-label learning problem into two classification sub-problems where each sub-problem corresponds to one label set of the original label space. Two multi-class classifiers are learned from each label set. In this way, two new classification sub-problems are independent from each other.

Generally, in multi-label learning, if the total number of labels is L, there may be up to 2^L label cases. If one wants to count all these cases and then decompose the original task into multiple multi-class problems, a huge computation cost is unavoidable. Fortunately, in dual set multi-label learning, there are up to $L_1 \times L_2$ label cases. *Co-Occurrence Based*

Algorithm 1 The DSML algorithm

Input: Training set $\mathcal{D} = \{(\boldsymbol{x}_i, y_i^a, y_i^b)|1 \leq i \leq m\}$, base learning algorithm $\mathcal{A}$, number of rounds T, weight tuning parameter B

Training process:

```
w^a_{1,i} = w^b_{1,i} = 1/m;
for t = 1 to T do
    (X^a_s, y^a_s) ← Sample(D, w^a_t)
    (X^b_s, y^b_s) ← Sample(D, w^b_t)
    Training three models h^raw_t, h^a_t and h^b_t with model-reuse mechanism by Eq. (1), (2) and (3)
    Calculating error rate ε^a_t and ε^b_t by Eq. (4) and (5)
    if ε^a_t > (L_1 − 1)/L_1 or ε^b_t > (L_2 − 1)/L_2 then
        Break
    end if
    Updating model weight α^a_t and α^b_t by Eq. (6) and (7)
    Updating sample distribution w^a_{t+1} and w^b_{t+1} by α^a_t, α^b_t and B with distribution adjusting mechanism according to Eq. (8) and (9)
    Performing normalization to w^a_{t+1} and w^b_{t+1}
end for
```

Output: Predict labels for dual set: $f^a(\boldsymbol{x})$ and $f^b(\boldsymbol{x})$ by Eq. (10) and (11)

Decomposition is raised based on this observation. It decomposes the task into a multi-class problem based on co-occurrence of labels. The method enumerates all the cases of label co-occurrence, and takes each case as a class label of the transformed multi-class problem.

Moreover, we can transform dual set multi-label learning problem into two multi-class classification problems where a task depends on the previous one. We call it the *Label Stacking* approach. Specifically, a multi-class classifier is trained on one label set, and the other multi-class classifier is trained on the other label set along with labels from the previous label set.

The DSML Approach

In this section, we propose a novel algorithm named DSML specifically designed for the dual set multi-label problem. As shown in Algorithm 1, DSML maintains the general outline of boosting. It decomposes the original problem into two dependent classification problems in the boosting framework. In this way, base classifier is responsible for dealing with the intra-set label relationship. At each boosting round, the models on two labels sets interact and help each other with the proposed model-reuse and distribution adjusting mechanisms, which could effectively exploit the inter-set label relationship.

Model-Reuse Mechanism Since DSML decomposes the original problem into two dependent sub-problems, only one sample distribution is not enough. Therefore, different from standard boosting approach designed for traditional supervised learning, during the training process of DSML, two sample distributions are maintained, one for each label set. In detail, in the t-th round of boosting, we have two m-dimensional sample distributions w_t^a and w_t^b, where the i-th value $w_{t,i}^a$ and $w_{t,i}^b$ is the weight for the i-th instance with respect to label set a and b, respectively. Then, two datasets (X_s^a, y_s^a) and (X_s^b, y_s^b) are sampled from the original training set according to the distribution w_t^a and w_t^b. Here X_s^a and X_s^b are instance matrices, and each row of them is an instance. y_s^a and y_s^b are label vectors associated with them. Inspired by (Huang, Yu, and Zhou 2012), we then propose a model-reuse mechanism to make two label sets help each other. Without loss of generality, we assume that $L_1 \geq L_2$, starting with the label set with fewer labels, i.e., we train a base classifier for label set b from the sampled set:

$$h_t^{raw} \leftarrow \mathcal{A}(X_s^b, y_s^b). \tag{1}$$

Then the model h^{raw} is used to make predictions for X_s^a, i.e., we have $\hat{Y^b} = h_t^{raw}(X_s^a)$. After that, this pseudo-label matrix is concatenated with the original features to form the new features for the task corresponding to label set a. That means the classifier on label set a is trained according to:

$$h_t^a \leftarrow \mathcal{A}([X_s^a, \hat{Y^b}], y_s^a). \tag{2}$$

Similarly, the predictions $\hat{Y^a} = h_t^a([X_s^b, Y_s^b])$ is concatenated with X_s^b to train the model for label set b:

$$h_t^b \leftarrow \mathcal{A}([X_s^b, \hat{Y^a}], y_s^b). \tag{3}$$

Here, $\hat{Y^a}$ and $\hat{Y^b}$ are 0-1 matrices, where each 1 indicates the instance is associated with th certain label, 0 otherwise.

In this way, the information embedded in the model of one label set can be reused by the other model, and they are expected to help each other improve their performance These steps are illustrated in Figure 2.

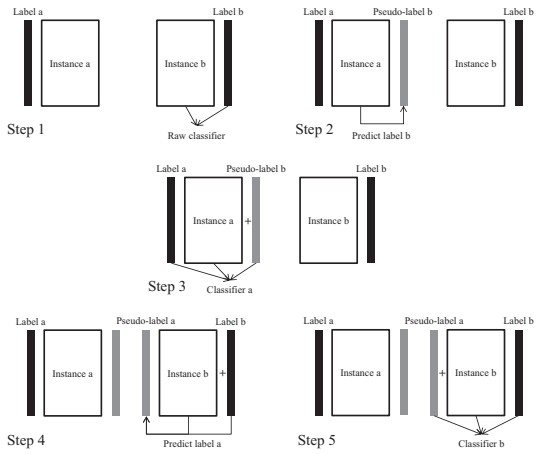


Figure 2: Five steps of model-reuse mechanism. Each white block represents the instance space. Each black strip refers to real labels and each gray strip stands for pseudo-label matrix predicted by learned classifiers.

Afterwards, the error rates ϵ_t^a and ϵ_t^b on each sample sets

are calculated by:

$$\epsilon_t^a = \sum_{i=1}^{m} [\![h_t^a([X_s^a, \hat{Y^b}]_i) \neq (y_s^a)_i]\!], \quad (4)$$

$$\epsilon_t^b = \sum_{i=1}^{m} [\![h_t^b([X_s^b, \hat{Y^a}]_i) \neq (y_s^b)_i]\!], \quad (5)$$

where $[\![\cdot]\!]$ is the indicator function which outputs 1 when $\cdot$ is true, 0 otherwise. And the model weights α_t^a and α_t^b are updated by:

$$\alpha_t^a = \frac{1}{L_1}\left[\log\frac{1-\epsilon_t^a}{\epsilon_t^a} + \log(L_1 - 1)\right], \quad (6)$$

$$\alpha_t^b = \frac{1}{L_2}\left[\log\frac{1-\epsilon_t^b}{\epsilon_t^b} + \log(L_2 - 1)\right]. \quad (7)$$

Here, L_1 and L_2 are used to make model weights fit the multi-class problem. The extra items $\log(L_1 - 1)$ and $\log(L_2 - 1)$ are not artificial (Zhu et al. 2006); they make the new algorithm equivalent to fitting a forward stage-wise addition model using a multi-class exponential loss function. And we put $1/L_1$ and $1/L_2$ in front of the $\log(\cdot)$ operation to make the training process smoother. It is worth noting that when $L_1 = L_2 = 2$, the weights for models are identical to that of standard AdaBoost algorithm. If $\epsilon_t^a > (L_1 - 1)/L_1$ or $\epsilon_t^b > (L_2 - 1)/L_2$, the classifier h_t^a or h_t^b is considered too weak to get a better performance in the boosting framework, and the algorithm stops.

Distribution Adjusting Mechanism To better exploit the inter-set label relationship, we further propose a new mechanism to adjust the distribution of training data for each label set. In the classical AdaBoost algorithm, an instance on which the model has made mistake will be emphasized by assigning a higher weight. In our setting, if an instance $\boldsymbol{x}_i$ was misclassified by the model on label set a, then we also increase the weight w_i^b. In this way, the instance $\boldsymbol{x}_i$ will be emphasized when training the model on label set b, and it is expected that the model on set b can provide more accurate information about $\boldsymbol{x}_i$ in the next round. As a result, the model on set a will get additional help information from set b and may avoid the mistake on $\boldsymbol{x}_i$.

Based on the above motivation, the sample distribution w_{t+1}^a and w_{t+1}^b are updated according to:

$$w_{t+1,i}^a = w_{t,i}^a \exp(\alpha_t^a \cdot [\![y_i^a \neq \hat{y}_i^a]\!])(B \cdot [\![y_i^b \neq \hat{y}_i^b]\!]), \quad (8)$$

$$w_{t+1,i}^b = w_{t,i}^b \exp(\alpha_t^b \cdot [\![y_i^b \neq \hat{y}_i^b]\!])(B \cdot [\![y_i^a \neq \hat{y}_i^a]\!]), \quad (9)$$

where $\hat{y}_i^a$ and $\hat{y}_i^b$ are predicted by h^a and h^b, respectively. The items $\exp(\alpha_t^a \cdot [\![y_i^a \neq \hat{y}_i^a]\!])$ and $\exp(\alpha_t^b \cdot [\![y_i^b \neq \hat{y}_i^b]\!])$ mean that only the weights of instances misclassified increase while the other weights remain the same as before. $B \geq 1$ is the distribution adjusting parameter to increase the weight of an instance on one label set which is misclassified on the other label set. At the end of training process, the sample distributions w_{t+1}^a and w_{t+1}^b are normalized to form a valid distribution.

During the testing phase, labels are predicted for instance $\boldsymbol{x}$ according to:

$$f^a(\boldsymbol{x}) = \operatorname*{argmax}_{l_1} \sum_{t=1}^{T} \alpha_t^a \cdot [\![h_t^a([\boldsymbol{x}, h_t^{raw}(\boldsymbol{x})]) = l_1]\!], \quad (10)$$

$$f^b(\boldsymbol{x}) = \operatorname*{argmax}_{l_2} \sum_{t=1}^{T} \alpha_t^b \cdot [\![h_t^b([\boldsymbol{x}, h_t^a([\boldsymbol{x}, h_t^{raw}(\boldsymbol{x})])]) = l_2]\!], \quad (11)$$

where $l_1 = 1, \cdots, L_1$ and $l_2 = 1, \cdots, L_2$. Obviously, the model-reuse mechanism is employed again just like that in the training phase.

Theoretical Results

In this section, we provide some theoretical analyses for the dual set multi-label learning, in particular, we are interested in the effect of splitting the total label set into dual sets. Due to the page limits, some preliminary definitions and proofs for theorems are omitted, which will be provided in a longer version.

Theorem 1. *For dual set multi-label learning problems, h^a and h^b are classifiers trained on the instance space $\mathcal{X}$ and label space $\mathcal{Y}^a$, $\mathcal{Y}^b$ respectively. h is a classifier trained directly from $\mathcal{X} \times [\mathcal{Y}^a \times \mathcal{Y}^b]$, namely,*

$$h : \boldsymbol{x} \to \operatorname*{arg\,max}_{y^a, y^b \in [\mathcal{Y}^a \times \mathcal{Y}^b]} h(\boldsymbol{x}, y),$$

where $y = [y^a, y^b]$, then margin of learning from dual label set is larger than that of directly learning from all labels:

$$\min\{\bar{\rho}_{h^a}(\boldsymbol{x}, y^a), \bar{\rho}_{h^b}(\boldsymbol{x}, y^b)\} \geq \bar{\bar{\rho}}_h(\boldsymbol{x}, y),$$

where $\bar{\rho}$ is margin for multi-class classification defined in (Mohri, Rostamizadeh, and Talwalkar 2012), and $\bar{\bar{\rho}}$ is defined as,

$$\bar{\bar{\rho}}_h(\boldsymbol{x}, y) = \min\{g(\boldsymbol{x}, y^a), g(\boldsymbol{x}, y^b)\} - \max_{y' \neq y^a \wedge y' \neq y^b} g(\boldsymbol{x}, y').$$

Remark. From Theorem 1, we can see that the margin of h is bounded by the minimum of margin of h^a and h^b. The margin is the larger the better. Thus, this bound implies the effectiveness of splitting the whole label set into two disjoint label sets. This exactly accords with our intuition, that we should consider the hierarchical structure in label sets.

Consider the approach that splits label sets into dual sets, we name it as splitting approach:

$$h^{spl}(\boldsymbol{x}) = [h^a(\boldsymbol{x}), h^b(\boldsymbol{x})],$$

then we give the definitions of empirical margin loss and risks based on *hamming loss* as follows:

Definition 2. *(Empirical Margin Loss (Mohri, Rostamizadeh, and Talwalkar 2012))*

$$\hat{R}_\rho(h) = \frac{1}{m}\sum_{i=1}^{m} \Phi_\rho(\rho_h(\boldsymbol{x}_i, y_i)),$$

where $\Phi_\rho(\cdot)$ is the margin loss function defined as:

$$\Phi_\rho = \begin{cases} 0, & \text{if } \rho \leq x \\ 1 - x/\rho, & \text{if } 0 \leq x \leq \rho \\ 1. & \text{if } x \leq 0 \end{cases}$$

Remark. Since margin loss function is a monotonously non-increasing function, it means that the larger margin is, the less loss will be.

Definition 3. *(Risks Based on Hamming Loss)*

$$R(h) = \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\left[\frac{1}{L_1+L_2}\sum_{\ell=1}^{L_1+L_2}[\![h_\ell(\boldsymbol{x}) \neq y_\ell]\!]\right],$$

$$R(h^a) = \mathbb{E}_{(\boldsymbol{x},y^a)\sim\mathcal{D}}\left[\frac{1}{L_1}\sum_{\ell=1}^{L_1}[\![h_\ell^a(\boldsymbol{x}) \neq y_\ell^a]\!]\right],$$

$$R(h^b) = \mathbb{E}_{(\boldsymbol{x},y^b)\sim\mathcal{D}}\left[\frac{1}{L_2}\sum_{\ell=1}^{L_2}[\![h_\ell^b(\boldsymbol{x}) \neq y_\ell^b]\!]\right].$$

Observation. The losses of these approaches satisfy,

$$[\![h_\ell(\boldsymbol{x}) \neq y_\ell]\!] \leq \max\{[\![h_\ell^a(\boldsymbol{x}) \neq y_\ell^a]\!], [\![h_\ell^b(\boldsymbol{x}) \neq y_\ell^b]\!]\}.$$

Proof. Since $[\![\cdot]\!]$ is either 1 or 0, we only need to bound the case when the right hand side is equal to 0.

As we know that $h(\boldsymbol{x}) = [h^a(\boldsymbol{x}), h^b(\boldsymbol{x})]$ and $y = [y^a, y^b]$, when $[\![h_\ell^a(\boldsymbol{x}) \neq y_\ell^a]\!] = 0 \wedge [\![h_\ell^b(\boldsymbol{x}) \neq y_\ell^b]\!] = 0$, we have left hand side as $[\![h_\ell(\boldsymbol{x}) \neq y_\ell]\!] = 0$. □

Based on Definition 2 and 3, we have the following generalization bound of the approach that splits the total label set into dual label sets:

Theorem 2. *Let $H = \{(\boldsymbol{x}, y^a, y^b) \in \mathcal{X} \times [\mathcal{Y}^a \times \mathcal{Y}^b] \to \mathbf{w}^{\mathrm{T}}\phi(\boldsymbol{x}) | \sum_{\ell=1}^{L_1+L_2}\|\mathbf{w}\|_{\mathbb{H}}^2 \leq \Lambda^2\}$ be a hypothesis set with $y^a = 1, \cdots, L_1, y^b = 1, \cdots, L_2$, where $\phi : \mathcal{X} \to \mathbb{H}$ is a feature mapping induced by some positive definite kernel κ. Assume that $S \subset \{\boldsymbol{x} : \kappa(\boldsymbol{x}, \boldsymbol{x}) \leq r^2\}$, and fix $\rho > 0$, then for any $\delta > 0$, with probability at least $1-\delta$, the following generalization bound holds for all $h^{spl} = [h^a, h^b] \in H$:*

$$R(h^{spl}) \leq \hat{R}_\rho(h^{spl}) + \frac{2r\Lambda}{\rho}\sqrt{\frac{\max\{L_1, L_2\}}{m}} + 3\sqrt{\frac{\log(2/\delta)}{m}}.$$

Remark. From Theorem 2, we can see that it makes sense to split label sets to deal with dual-set multi-label learning since the convergence rate of generalization error is standard as $O(1/\sqrt{m})$. Besides, the error bound exhibits a radical dependence on the maximal number of labels in dual sets. This also implies a relatively balanced splitting on the label sets may improve the performance.

Table 1: Statistics of the datasets, where M, D, L_1, and L_2 denote the number of instances, dimensions, size of first label set, and size of second label set in each dataset, respectively.

Dataset	M	D	L_1	L_2
Calligrapher-Font	23195	512	14	5
Brand-Type	2247	4096	7	3
Frequency-Gender	3157	19	5	2

Experiments

Experimental Settings

Datasets We manually collect two real-world datasets and adapt one publicly available dataset for dual set multi-label learning. Details of them can be found in a longer version. Here we summarize their statistics in Table 1.

Evaluation Measures In dual set multi-label learning, we care about the performance on each individual set of labels as well as the overall performance, so we can evaluate the performance of compared algorithms with accuracy. Formally, we can define three kinds of accuracies as follows:

Definition 4. *Let $\mathcal{Z} = \{\boldsymbol{z}_i, y_i^a, y_i^b | 1 \leq i \leq n\}$ denote the testing set where n is the total number of testing instances and let h^a, h^b be the underlying classifiers learned from the training process associated with two label sets respectively. Three accuracies are defined to evaluate the performance,*

$$Accuracy_a = \frac{1}{n}\sum_{i=1}^{n}[\![h^a(\boldsymbol{z}_i) = y_i^a]\!],$$

$$Accuracy_b = \frac{1}{n}\sum_{i=1}^{n}[\![h^b(\boldsymbol{z}_i) = y_i^b]\!],$$

$$Accuracy_{all} = \frac{1}{n}\sum_{i=1}^{n}[\![h^a(\boldsymbol{z}_i) = y_i^a]\!] \cdot [\![h^b(\boldsymbol{z}_i) = y_i^b]\!].$$

In words, $Accuracy_a$ and $Accuracy_b$ evaluate the performance on the first and second label set, respectively. $Accuracy_{all}$ measures the overall performance.

Compared Methods In this part, all algorithms are evaluated on the same five-fold partition of the same datasets. Benchmark approaches are evaluated and compared. For Independent Decomposition, Co-Occurrence Based Decomposition, Label Stacking, and DSML, radial basis function (RBF) neural networks are used as their base classifiers with same hyper-parameters. For DSML, the number of boosting rounds T is set to 10, and B is set to 1.05.

Moreover, since dual set multi-label learning is a specific case of general multi-label learning, traditional multi-label algorithms can be used for this case. Four of these algorithms are compared, which are ML-KNN (Zhang and Zhou 2007), ML-RBF (Zhang 2009), BP-MLL (Zhang and Zhou 2006), and RankSVM (Elisseeff and Weston 2002). Due to the different settings of dual set multi-label learning and traditional multi-label learning, a little modification should be done to change the output of multi-label learning to fit dual set multi-label learning, which is firstly dividing labels into two sets and then choosing the label with the highest probability within each set as the final prediction label. For these methods, hyper-parameters are set according to the suggestions given by their papers.

Experimental Results

Algorithms Comparison Table 2 gives the five-fold cross-validation performance of all compared algorithms on the datasets. No accuracy on individual label set of the Co-Occurrence Based Decomposition is shown, because after

Table 2: The five-fold cross-validation performance of each compared algorithm (mean ± std.) on *Calligrapher-Font*, *Brand-Type*, and *Frequency-Gender* datasets. Best accuracies on each dataset are marked in bold font. N/A indicates no result of a certain block.

Dataset	Measure	Algorithms							
		DSML	Ind. Dec.	Co-Occ. Dec.	Label Stacking	ML-KNN	ML-RBF	BP-MLL	RankSVM
Cal.-Font	$Accy._a$	**.6562 ± .0059**	.5967 ± .0082	N/A	.6019 ± .0088	.6337 ± .0075	.6372 ± .0045	.1493 ± .0051	N/A
	$Accy._b$	**.7223 ± .0079**	.6751 ± .0040	N/A	.6801 ± .0078	.7101 ± .0030	.7100 ± .0087	.4104 ± .0670	N/A
	$Accy._{all}$	**.5672 ± .0087**	.4836 ± .0099	.5609 ± .0050	.4889 ± .0094	.5570 ± .0048	.5396 ± .0066	.0764 ± .0077	N/A
Brand-Type	$Accy._a$	.5723 ± .0226	.5661 ± .0129	N/A	**.5968 ± .0254**	.4722 ± .0160	.5207 ± .0223	.1206 ± .0182	.5238 ± .0352
	$Accy._b$	**.7730 ± .0249**	.7677 ± .0092	N/A	.7637 ± .0225	.7245 ± .0115	.7405 ± .0126	.3000 ± .0509	.7517 ± .0137
	$Accy._{all}$	**.4949 ± .0227**	.4744 ± .0105	.4784 ± .0294	.4735 ± .0302	.3912 ± .0078	.4201 ± .0160	.0538 ± .0053	.4183 ± .0345
Freq.-Gndr.	$Accy._a$	**.8521 ± .0091**	.8321 ± .0212	N/A	.8375 ± .0170	.5879 ± .0091	.7570 ± .0144	.4004 ± .1464	.0326 ± .0135
	$Accy._b$	.9547 ± .0061	.9579 ± .0067	N/A	.9550 ± .0051	.6953 ± .0196	**.9661 ± .0047**	.5014 ± .0271	.5382 ± .0643
	$Accy._{all}$	**.8220 ± .0082**	.8039 ± .0214	.8068 ± .0187	.8096 ± .0183	.4587 ± .0161	.7387 ± .0134	.1704 ± .0847	.0127 ± .0116

Table 3: The five-fold cross-validation performance of DSML (mean ± std.) on *Calligrapher-Font*, *Brand-Type*, and *Frequency-Gender* datasets when B increases from 1.00 to 1.20 with T fixed at 10. Best accuracies on each dataset are marked in bold font.

Dataset	Measure	Distribution Adjusting Parameter B							
		1.00	1.01	1.02	1.03	1.05	1.10	1.15	1.20
Cal.-Font	$Accy._a$	.6536 ± .0054	**.6576 ± .0064**	.6567 ± .0051	.6557 ± .0067	.6562 ± .0059	.6541 ± .0033	.6546 ± .0076	.6528 ± .0060
	$Accy._b$	.7225 ± .0060	.7244 ± .0062	.7249 ± .0043	**.7263 ± .0046**	.7223 ± .0079	.7246 ± .0041	.7210 ± .0037	.7230 ± .0054
	$Accy._{all}$	.5656 ± .0078	.5697 ± .0062	.5674 ± .0043	.5690 ± .0058	.5672 ± .0087	**.5698 ± .0043**	.5659 ± .0078	.5660 ± .0045
Brand-Type	$Accy._a$	.5710 ± .0296	.5657 ± .0259	.5706 ± .0303	.5706 ± .0206	**.5723 ± .0226**	.5648 ± .0185	.5710 ± .0201	.5603 ± .0343
	$Accy._b$	.7784 ± .0142	.7668 ± .0185	.7659 ± .0193	.7650 ± .0212	.7730 ± .0249	.7641 ± .0107	**.7788 ± .0182**	.7699 ± .0182
	$Accy._{all}$	.4905 ± .0324	.4847 ± .0227	.4856 ± .0257	.4882 ± .0231	**.4949 ± .0227**	.4824 ± .0073	.4922 ± .0228	.4833 ± .0340
Freq.-Gndr.	$Accy._a$	.8413 ± .0110	.8432 ± .0107	.8432 ± .0177	.8413 ± .0140	**.8521 ± .0091**	.8435 ± .0137	.8473 ± .0162	.8476 ± .0119
	$Accy._b$	.9541 ± .0071	.9531 ± .0041	.9512 ± .0073	.9554 ± .0074	.9547 ± .0061	.9515 ± .0040	.9557 ± .0054	**.9560 ± .0038**
	$Accy._{all}$	.8131 ± .0060	.8134 ± .0118	.8134 ± .0158	.8119 ± .0166	**.8220 ± .0082**	.8128 ± .0151	.8172 ± .0153	.8175 ± .0155

label co-occurrence counting, all dual labels are transformed into new multi-class labels. Also, the result of RankSVM is absent on *Calligrapher-Font* dataset, because no result is obtained after 10 times the running time of DSML.

From Table 2, we can see that DSML is significantly better than the other algorithms. It achieves the best overall accuracies on all datasets. On the largest dataset *Calligrapher-Font*, DSML is the best on all of the three criteria. It is worth noting that DSML performs much better than Independent Decomposition, which shows the effectiveness of boosting, model-reuse and distribution adjusting mechanisms. On *Brand-Type* dataset, DSML only loses to Label Stacking on the accuracy of the first label set. On *Frequency-Gender* dataset, DSML only loses to ML-RBF with a small gap on the accuracy of the second label set.

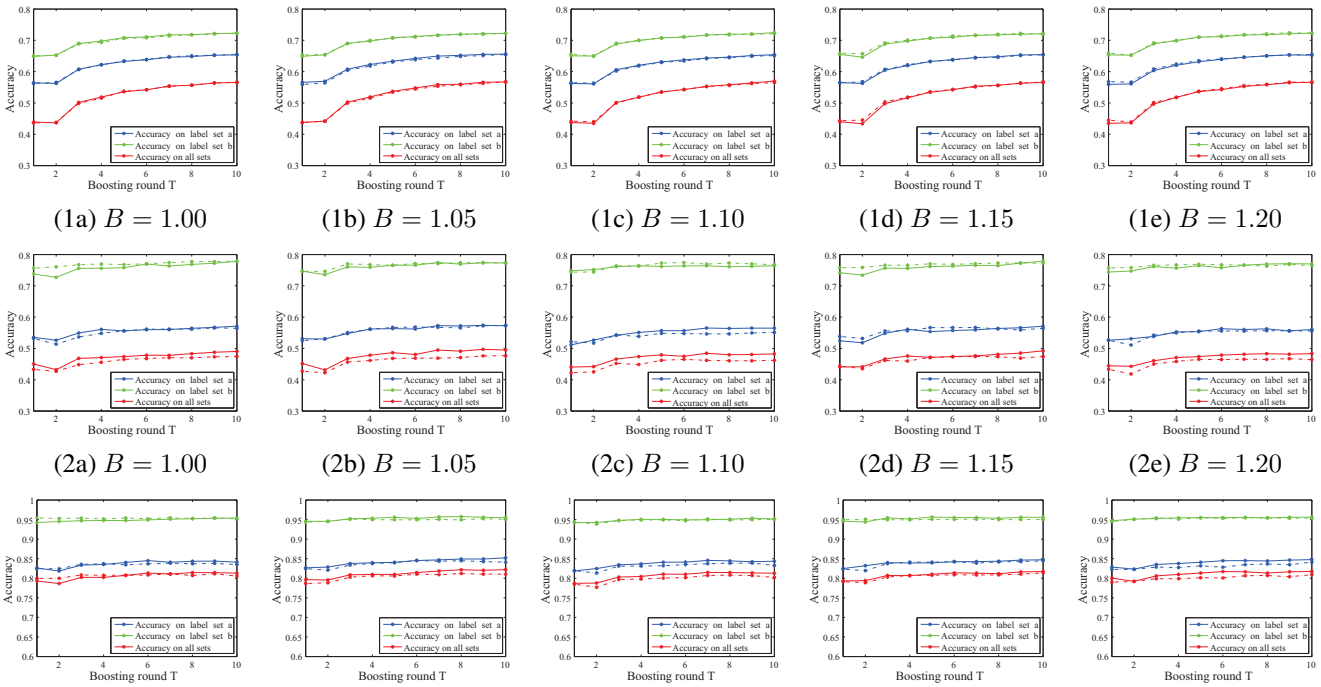
Among Independent Decomposition, Co-Occurrence Based Decomposition and Label Stacking, Independent Decomposition performs the worst on all datasets. By contrast, Co-Occurrence Based Decomposition performs better than Independent Decomposition, which validates that utilizing inter-set label relationship is helpful. However, it exploits this relationship in a rough way, which leads to different ranges of improvement on three datasets. Label co-occurrence relationship is more significant for *Calligrapher-Font* and *Brand-Type* datasets. But on *Frequency-Gender* dataset, whose label relationship mainly lies in label distribution rather than co-occurrence, the improvement is not significant. For Label Stacking, it has a similar performance to Independent Decomposition over the second label set, nevertheless, its accuracy is improved with respect to the first label set. Thus, it has a better overall accuracy. From these results, we can know that utilizing both inter-set and intra-set label relationships are important in dual set multi-label learning.

Among multi-label learning approaches, ML-RBF performs better than other methods except ML-KNN on the overall accuracy on *Calligrapher-Font* dataset. This is probably because that ML-KNN tends to perform better on large dataset, where more instances can be compared. BP-MLL performs the worst of all these approaches. Performance of RankSVM is relatively good on *Brand-Type* dataset, but it performs poorly on *Frequency-Gender* dataset, especially on the accuracy on the first label set and the overall accuracy. All these approaches perform worse than DSML, which proves that treating all labels equally is not an appropriate way to address dual set multi-label learning.

Study on Model-Reuse Mechanism In order to show the effectiveness of model-reuse mechanism, we perform experiments of DSML with and without model-reuse mechanism on all datasets. Results are given in Figure 3. T is set to be 10, with every round reported, and B is set between 1.00 and 1.20 with an interval of 0.05.

Obviously, DSML with model-reuse mechanism signifi-

Figure 3: The five-fold cross-validation performance of DSML (mean) on *Calligrapher-Font* (sub-figures (1a) to (1e)), *Brand-Type* (sub-figures (2a) to (2e)), and *Frequency-Gender* (sub-figures (3a) to (3e)) datasets change with and without model-reuse mechanism when T increases with fixed value of B. The solid lines represent the performance with model-reuse mechanism, dotted lines otherwise.



(1a) $B = 1.00$ (1b) $B = 1.05$ (1c) $B = 1.10$ (1d) $B = 1.15$ (1e) $B = 1.20$

(2a) $B = 1.00$ (2b) $B = 1.05$ (2c) $B = 1.10$ (2d) $B = 1.15$ (2e) $B = 1.20$

(3a) $B = 1.00$ (3b) $B = 1.05$ (3c) $B = 1.10$ (3d) $B = 1.15$ (3e) $B = 1.20$

cantly outperforms the one without it, which validates that model-reuse mechanism plays a key role in improving the performance of DSML. On *Brand-Type* and *Frequency-Gender* datasets, though some green solid lines are below the dotted ones, the improvement of overall accuracies is significant, with most red solid lines above the dotted ones.

In addition, from Figure 3, we can observe that when B is fixed, the performance of DSML is unstable in the initial increasing phase of T, especially when $T = 2$. After that, DSML improves remarkably especially on *Calligrapher-Font* dataset. Then, the performance of DSML tends to be stable in the remaining increasing phase of T. This phenomenon accords with our intuition since DSML is an approach with a boosting framework.

Study on Distribution Adjusting Mechanism Since B controls the effect of distribution adjusting mechanism, in order to illustrate the positive influence of it, we perform experiments of DSML with different B settings. It is worth mentioning that $B = 1.00$ means DSML performs without distribution adjusting mechanism. Table 3 reports how DSML performs on all different datasets with five-fold cross-validation as B increases from 1.00 to 1.20. The boosting round T is fixed at 10.

We can observe that the better performance can be achieved when B is larger than 1.00 over all datasets, which validates that adjusting the distribution according to the information from the other label set can improve the performance. It also implies the importance of exploiting the inter-set label relationship. In practice, we can see that smaller or larger B does not improve the performance very much. On all datasets used in this paper, we find that $B = 1.05$ or 1.10 may be a relatively proper setting.

Conclusion

In this paper, we propose a novel learning framework named dual set multi-label learning, where an object is associated with two labels, each of which comes from one of the dual label sets. We also propose a boosting style algorithm to solve this problem. On one hand, a base classifier is used to utilize the exclusive relationship among labels within the same set; on the other hand, model from each label set is reused by the other one, and data distributions are jointly adjusted such that the mistakes on one model will be made up by the other one. Moreover, theoretical analyses are presented to show the superiority of learning from dual label sets to learning directly from all labels. Experimental studies on three real-world datasets validate the effectiveness of our proposed approach. It is worth noting that since model-reuse mechanism plays a key role in DSML, it can be extended to problems with multiple label sets. As a boosting style approach, DSML can be more powerful with stronger base classifiers. In the future, more applications will be studied under the framework of dual set multi-label learning and it will be extended to multiple label sets cases.

References

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

Cesa-Bianchi, N.; Gentile, C.; and Zaniboni, L. 2006. Hierarchical classification: combining bayes with svm. In *Proceedings of the 23rd International Conference on Machine Learning*, 177–184.

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

Freund, Y., and Schapire, R. E. 1995. A decision-theoretic generalization of on-line learning and an application to boosting. In *Proceedings of the 2nd European Conference on Computational Learning Theory*, 119–139.

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.

Hariharan, B.; Zelnik-Manor, L.; Vishwanathan, S. V. N.; and Varma, M. 2010. Large scale max-margin multi-label classification with priors. In *Proceedings of the 27th International Conference on Machine Learning*, 423–430.

Huang, S.-J., and Zhou, Z.-H. 2012. Multi-label learning by exploiting label correlations locally. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, 949–955.

Huang, S.-J.; Yu, Y.; and Zhou, Z.-H. 2012. Multi-label hypothesis reuse. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 525–533.

Liu, W., and Tsang, I. W. 2015. On the optimality of classifier chain for multi-label classification. In *Advance in Neural Information Processing Systems 28*, 712–720.

Liu, W., and Tsang, I. W. 2017. Making decision trees feasible in ultrahigh feature and label dimensions. *Journal of Machine Learning Research* 18(81):1–36.

Liu, W.; Tsang, I. W.; and Müller, K.-R. 2017. An easy-to-hard learning paradigm for multiple classes and multiple labels. *Journal of Machine Learning Research* 18(94):1–38.

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

Mohri, M.; Rostamizadeh, A.; and Talwalkar, A. 2012. *Foundations of machine learning*. MIT press.

Rauber, T. W.; Mello, L. H.; Rocha, V. F.; Luchi, D.; and Varejão, F. M. 2014. Recursive dependent binary relevance model for multi-label classification. In *Proceedings of the 14th Ibero-American Conference on Artificial Intelligence*, 206–217.

Read, J.; Pfahringer, B.; Holmes, G.; and Frank, E. 2011. Classifier chains for multi-label classification. *Machine Learning* 85(3):254–269.

Rousu, J.; Saunders, C.; Szedmak, S.; and Shawe-Taylor, J. 2005. Learning hierarchical multi-category text classification models. In *Proceedings of the 22nd International Conference on Machine Learning*, 744–751.

Schapire, R. E., and Singer, Y. 1999. Improved boosting algorithms using confidence-rated predictions. *Machine Learning* 37(3):297–336.

Tsoumakas, G.; Dimou, A.; Spyromitros, E.; Mezaris, V.; Kompatsiaris, I.; and Vlahavas, I. 2009. Correlation-based pruning of stacked binary relevance models for multi-label learning. In *Proceedings of the 1st International Workshop on Learning from Multi-Label Data*, 101–116.

Wang, J.; Yang, Y.; Mao, J.; Huang, Z.; Huang, C.; and Xu, W. 2016. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the 29th IEEE Conference on Computer Vision and Pattern Recognition*, 2285–2294.

Yeh, C. K.; Wu, W. C.; Ko, W. J.; and Wang, Y. C. F. 2017. Learning deep latent spaces for multi-label classification. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, 2838–2844.

Zhang, M.-L., and Zhou, Z.-H. 2006. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE Transactions on Knowledge & 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.

Zhang, M.-L., and Zhou, Z.-H. 2014. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge & Data Engineering* 26(8):1819–1837.

Zhang, M.-L. 2009. ML-RBF: RBF neural networks for multi-label learning. *Neural Processing Letters* 29(2):61–74.

Zhou, Z.-H. 2012. *Ensemble Methods: Foundations and Algorithms*. Taylor & Francis.

Zhu, J.; Zou, H.; Rosset, S.; and Hastie, T. 2006. Multi-class adaboost. *Statistics & Its Interface* 2(3):349–360.