Multi-Label Learning with Weak Label

Yu-Yin Sun   Yin Zhang   Zhi-Hua Zhou[*]
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210093, China
{sunyy, zhangyin, zhouzh}@lamda.nju.edu.cn

Abstract

Multi-label learning deals with data associated with multiple labels simultaneously. Previous work on multi-label learning assumes that for each instance, the "full" label set associated with each training instance is given by users. In many applications, however, to get the full label set for each instance is difficult and only a "partial" set of labels is available. In such cases, the appearance of a label means that the instance is associated with this label, while the absence of a label does not imply that this label is not proper for the instance. We call this kind of problem "weak label" problem. In this paper, we propose the WELL (WEak Label Learning) method to solve the weak label problem. We consider that the classification boundary for each label should go across low density regions, and that each label generally has much smaller number of positive examples than negative examples. The objective is formulated as a convex optimization problem which can be solved efficiently. Moreover, we exploit the correlation between labels by assuming that there is a group of low-rank *base* similarities, and the appropriate similarities between instances for different labels can be derived from these base similarities. Experiments validate the performance of WELL.

Introduction

In traditional supervised learning, each instance is associated with *one* label that indicates its concept class belongingness. In many real-world problems, however, one object usually belongs to multiple concepts simultaneously. For example, in text categorization, a document on national health service belongs to several predefined topics such as *government* and *health* simultaneously; in image or video annotation, an image showing a tiger in woods is associated with several annotated words such as *tiger* and *trees* simultaneously. One label per instance is out of its capability for dealing with such scenario, and *multi-label learning* has thus attracted much attention. Under the framework of multi-label learning, each instance is associated with multiple labels, indicating the concepts it belongs to.

In previous multi-label studies, a basic assumption is that all the proper labels of every training instance are given. For example, if a training image contains the concepts *tiger*, *trees* and *forest*, the user should provide the labels *tiger*, *trees* and *forest* for the image. In many applications, however, this assumption hardly holds since getting all the proper labels is difficult, and generally only a "partial" label set is available. For example, the user may only tag the image with the label *tiger* while missing the labels *trees* and *forest*. In such scenario, if the user provides a label for the instance, we know that this is a proper label for this instance; while for a label which has not been assigned to the instance, we could not conclude that this is not a proper label for the instance. It is evident that this scenario is quite different from the classic multi-label setting where all proper labels for training instances are assumed to be given. We call this kind of multi-label problem the "weak label" problem.

The weak label problem is related to but different from the PU-learning (Positive and Unlabeled data Learning) problem (Li and Liu 2003; Liu et al. 2003; Fung et al. 2006; Elkan and Noto 2008). If all labels are independent, a weak label problem can be decomposed into a series of PU-learning problems, each corresponding to a label. Such simple decomposition, however, ignores the correlation between the labels that can be very useful. For example, the appearance of the label *computer* in an image strongly implies the existence of the label *desk* and the nonexistence of the label *tiger*. In weak label problem, since the label information of training examples is incomplete, we may want to exploit the label correlation rather than simply treating the labels independently. Moreover, PU-learning methods generally do not consider class imbalance, while class imbalance inherently exists in the weak label problem because there are multiple labels and for each label, the number of positive examples is usually much smaller than the number of negative examples.

In this paper, we study the weak label problem and propose the WELL (WEak Label Learning) method. We require the classification boundary for each label to go across low density regions, and explicitly consider the inherent class imbalance in the weak label problem. We formulate the objective as a convex optimization problem which can be solved by quadratic programming efficiently. To exploit the label correlation, we assume that there is a group of low-rank

[*] This research was supported by the National Fundamental Research Program of China (2010CB327903), the National Science Foundation of China (60975043, 60721002), the Jiangsu 333 High-Level Talent Cultivation Program and the MSRA IST Program.

base similarities, and for each label, an appropriate similarity between instances can be derived from these base similarities. By "appropriate similarity" we mean that, instances that are similar according to this similarity tend to have the same belongingness of the concerned label, and vice versa. The superior performance of the proposed WELL method is validated in experiments.

The rest of the paper is organized as follows. We start by a brief review of related work. Then, we formulate the weak label problem and propose the WELL method. A variety of experiments are reported, followed by the conclusion.

Related Work

A straightforward approach to multi-label learning is to decompose the task into a number of binary classification problems, each for one label (Joachims 1998; Yang 1999; Boutell et al. 2004). Such simple approach would encounter many difficulties, among which is the inherent class imbalance of multi-label problem, that is, the number of positive examples for each label is usually much smaller than that of negative examples. There were some efforts for relaxing the problem caused by class imbalance. For example, Zhang and Zhou (2007) considered label prior probabilities gained from the k-nearest neighbors of the instance and utilized *maximum a posteriori* (MAP) principle to determine proper labels in their ML-kNN method. Another difficulty is on the exploitation of the correlation among class labels. Many multi-label learning methods have tried to consider the label correlation in different ways. Examples include methods based on probabilistic generative models (McCallum 1999; Ueda and Saito 2003), maximum entropy methods (Ghamrawi and McCallum 2005; Zhu et al. 2005), hypergraph spectral learning (Sun, Ji, and Ye 2008), shared subspace classification (Ji et al. 2008), models-shared subspace boosting (Yan, Těsić, and Smith 2007), maximizing the dependence between features and labels (Zhang and Zhou 2010), etc. Some multi-label learning methods work by transforming the task into a ranking problem, trying to rank the proper labels before other labels for each instance. Representative methods include BoosTexter (Schapire and Singer 2000), RankSVM (Elisseeff and Weston 2002), etc.

PU-learning, also known as partially supervised learning (Li and Liu 2003), studies the problem where a small positive example set (P) and a large unlabeled example set (U) are given for training. This is a special kind of semi-supervised learning where there is no labeled negative examples. Many existing PU-learning methods (Li and Liu 2003; Liu et al. 2003; Fung et al. 2006) first try to obtain a set of labeled negative examples, by considering the instances which are with the highest confidence to be negative, and then train a supervised or semi-supervised classifier. Some methods (Lee and Liu 2003; Liu et al. 2003) treat all the unlabeled instances as negative and assign different costs or weights to different kinds of errors, where the costs associated with labeled data are larger than those associated with unlabeled data. Based on the assumption that the labeled positive examples are sampled completely randomly from all the potential positive examples, Elkan and Noto (2008) showed that a PU-learner predicts probabilities that differ by only a constant factor from the true conditional probabilities of being positive, and this factor can be estimated on validation sets.

The WELL Method

Let $\mathcal{X}$ denotes the feature space and suppose there is a label set Θ containing m different labels. The proper labels associated with an instance $\boldsymbol{x} \in \mathcal{X}$ compose a subset of Θ, which can be represented as an m-dimensional binary label vector, with 1 indicating that the instance belongs to the concept corresponding to the dimension and 0 otherwise. All the labels consist of the label space $\mathcal{Y} = \{0,1\}^m$. In the classic multi-label learning setting, for n labeled instances we have a full label matrix $Y \in \{0,1\}^{n \times m}$ where $Y_{ik} = 1$ means the k-th label is a proper label while $Y_{ik} = 0$ means the k-th label is not a proper label for the i-th instance. In the weak label problem, Y is unknown and instead we are given a partial label matrix $\hat{Y} \in \{0,1\}^{n \times m}$ where $\hat{Y}_{ik} \leq Y_{ik}$. Different from the full label matrix, $\hat{Y}_{ik} = 0$ tells us nothing. We want to learn a predicted label matrix $F \in \{0,1\}^{n \times m}$ from $\hat{Y}$ to approximate Y.

Problem Formulation

We observe that in most multi-label applications, for each label the number of positive examples is much smaller than that of negative examples. Therefore, we want the predicted positive instances to be as sparse as possible for each label, i.e., we want to minimize $\mathbf{1}^{\mathrm{T}} F_{\cdot k}$, where $\mathbf{1}$ is all-one column vector and $F_{\cdot k}$ is the k-th column of F. Similar to graph-based unsupervised/semi-supervised learning methods (Shi and Malik 2000; Belkin, Niyogi, and Sindhwani 2006), we construct a PSD (positive semi-definite) similarity matrix $W = [W_{ij}]_{n \times n}$, where W_{ij} is the similarity between the i-th and the j-th instances. Minimizing $\sum_{i,j} (F_{ik} - F_{jk})^2 W_{ij}$ is equivalent to requiring the classification boundary for each label to go across low density regions. Thus, the prediction of the k-th label, $F_{\cdot k}$, is the solution to the optimization problem

$$\begin{aligned} \min_{\boldsymbol{f}} \quad & \mathbf{1}^{\mathrm{T}} \boldsymbol{f} + \alpha \sum_{i,j} (f_i - f_j)^2 W_{ij} + \beta \sum_{\hat{Y}_{ik}=1} \ell(\hat{Y}_{ik}, f_i) \\ \text{s.t.} \quad & \boldsymbol{f} \in \{0,1\}^n \ , \end{aligned} \tag{1}$$

where α and β are the controlling parameters, $\ell(\cdot,\cdot)$ is the loss function occurring only on the location (i,k) where $\hat{Y}_{ik} = 1$. Solving Eq. 1 is hard and therefore, we relax the domain of $\boldsymbol{f}$ from $\{0,1\}^n$ to $[0,1]^n$. If we define $\ell(\cdot,\cdot)$ as a convex function, this problem is a convex optimization problem. Denote $D = \mathrm{diag}(d_1, d_2, \cdots, d_n)$ a diagonal matrix with diagonal elements $d_i = \sum_j W_{ij}$ and $L = D - W$ the Lapacian matrix. Adopting the squared loss for the loss function, we can transform Eq. 1 to the following QP (quadratic programming) problem

$$\begin{aligned} \min_{\boldsymbol{f}} \quad & \boldsymbol{f}^{\mathrm{T}} (\alpha L + \beta \Upsilon_k) \boldsymbol{f} - 2\big(\beta \hat{Y}_{\cdot k} + \mathbf{1}\big)^{\mathrm{T}} \boldsymbol{f} \\ \text{s.t.} \quad & \boldsymbol{f} \in [0,1]^n \ , \end{aligned} \tag{2}$$

where $\Upsilon_k = \mathrm{diag}(\hat{Y}_{1k}, \hat{Y}_{2k}, \cdots, \hat{Y}_{nk})$ is a diagonal matrix.

For each label, after obtaining the continuous $\boldsymbol{f}$, we get the ranking of instances. To get the prediction we need to learn a threshold t to discretize $\boldsymbol{f}$ as $F_{ik}=\delta(f_i \geq t)$, where δ is the indicator function which takes 1 when $f_i \geq t$ and 0 otherwise. Since we only have positive data, we could not adopt regression methods to train a model predicting t like in (Elisseeff and Weston 2002). Note that as t varies, $F_{\cdot k}$ can only take $n+1$ different values. So, a simple strategy is to choose the $F_{\cdot k}$ that minimizes Eq. 1.

Shared Base Similarities

In Eq. 1 all the labels share the same similarity matrix W derived from the feature space. In many applications, however, there is a gap between the similarity for features and the similarity for semantic meanings. Similarity which is helpful for the classification should be dependent on the corresponding label. Take image annotation task for example. Assume that we have three images, where the first image has labels *car* and *road*, the second has labels *people* and *road*, and the third has labels *people*, *computer* and *office*. When the label *road* is concerned, the second image is similar to the first one and dissimilar to the third one because it shares *road* with the first image while the third image has no *road*. While when the label *people* is concerned, the second image is similar to the third one and dissimilar to the first one. It is evident that it is better to learn different similarity matrices for different labels, while simply using the same similarity matrix may lose useful information.

In order to embed the label information, we learn the new similarity matrix for the k-th label, denoted as W^k, by maximizing the kernel alignment with $\hat{Y}_{\cdot k}$, i.e., $\hat{Y}_{\cdot k}^{\mathrm{T}} W^k \hat{Y}_{\cdot k} / \left\| W^k \right\|_F$. Note that when $\hat{Y}_{ik}=0$, we have $\hat{Y}_{ik} W_{ij}^k \hat{Y}_{jk}=0$ for any j, which means that the uncertainty of $\hat{Y}_{ik}$ will not contribute to the kernel alignment. In the meanwhile, similar to (Liu et al. 2009), we treat the original similarity matrix as a noisy observation of W^k and require W^k be in the neighborhood of W. Thus, we have

$$\begin{aligned} \min_{W^k} \quad & -\hat{Y}_{\cdot k}^{\mathrm{T}} W^k \hat{Y}_{\cdot k} + \gamma \left\| W^k - W \right\|_F^2 \\ \text{s.t.} \quad & \left\| W^k \right\|_F = C, \quad W^k \succeq 0 \;, \end{aligned} \tag{3}$$

where γ is the controlling parameter, C is a constant used to control the Frobenius norm of W^k, and $W^k \succeq 0$ means W^k is a PSD matrix. However, contrary to the extreme where all the labels share the same similarity matrix in Eq. 1, Eq. 3 goes to the other extreme where the label correlation is not taken into consideration. In order to get a balance, we require all the W^k's share something. Similar strategies have been used before. For example, in (Yan, Těsić, and Smith 2007), a group of base models is shared by all the models corresponding to different labels; in (Ji et al. 2008), a common subspace is shared by all the classification tasks. Here, we assume that there is a group of low-rank base similarities, and the appropriate similarities between instances for different labels can be derived from these base similarities, i.e., $W^k = \sum_i \lambda_i^k W_i$ where $W_i = \boldsymbol{v}_i \boldsymbol{v}_i^{\mathrm{T}}$ and $\boldsymbol{v}_i$ is the i-th orthonormal eigenvector of W as $W=\sum_i \eta_i \boldsymbol{v}_i \boldsymbol{v}_i^{\mathrm{T}}$. This is related to some studies in the MIML (multi-instance multi-label learning) framework where the high-level concept is derived from a set of sub-concepts (Zhou et al. 2008). To ensure $W^k \succeq 0$, we require $\lambda_i^k \geq 0$ for each k and i. Denote $\boldsymbol{\lambda}^k = (\lambda_1^k, \cdots, \lambda_n^k)$, we have $\|W^k\|_F = \|\boldsymbol{\lambda}^k\|_2$ and the constraint becomes $\|\boldsymbol{\lambda}^k\|_2 = C$. For simplicity, we define $C=\sum_i \eta_i$ and replace the ℓ_2-norm constraint on $\boldsymbol{\lambda}^k$ by the ℓ_1-norm constraint. Thus Eq. 3 becomes [1]

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & -\hat{Y}_{\cdot k}^{\mathrm{T}} \Big(\sum\nolimits_i \lambda_i W_i \Big) \hat{Y}_{\cdot k} + \gamma \Big\| \sum\nolimits_i \lambda_i W_i - W \Big\|_F^2 \\ \text{s.t.} \quad & \|\boldsymbol{\lambda}\|_1 = C, \quad \lambda_i \geq 0 \;. \end{aligned} \tag{4}$$

Note that

$$\begin{aligned} \hat{Y}_{\cdot k}^{\mathrm{T}} \Big(\sum\nolimits_i \lambda_i W_i \Big) \hat{Y}_{\cdot k} &= \sum\nolimits_i \lambda_i \Big(\hat{Y}_{\cdot k}^{\mathrm{T}} W_i \hat{Y}_{\cdot k} \Big) \\ &= \sum\nolimits_i \lambda_i \Big(\hat{Y}_{\cdot k}^{\mathrm{T}} \boldsymbol{v}_i \boldsymbol{v}_i^{\mathrm{T}} \hat{Y}_{\cdot k} \Big) = \boldsymbol{u}^{\mathrm{T}} \boldsymbol{\lambda} \;, \end{aligned}$$

where $u_i = \hat{Y}_{\cdot k}^{\mathrm{T}} \boldsymbol{v}_i \boldsymbol{v}_i^{\mathrm{T}} \hat{Y}_{\cdot k}$. We have [2]

$$\begin{aligned} & \Big\| \sum\nolimits_i \lambda_i W_i - W \Big\|_F^2 \\ = \; & \sum\nolimits_{i,j} \lambda_i \lambda_j \mathrm{tr}\,(W_i W_j) - 2 \sum\nolimits_i \lambda_i \mathrm{tr}\,(W_i W) \\ = \; & \sum_{i,j} \lambda_i \lambda_j \boldsymbol{v}_i^{\mathrm{T}} \boldsymbol{v}_j \boldsymbol{v}_j^{\mathrm{T}} \boldsymbol{v}_i - 2 \sum_{i,j} \lambda_i \eta_j \boldsymbol{v}_i^{\mathrm{T}} \boldsymbol{v}_j \boldsymbol{v}_j^{\mathrm{T}} \boldsymbol{v}_i \\ = \; & \boldsymbol{\lambda}^{\mathrm{T}} I \boldsymbol{\lambda} - 2 \boldsymbol{\eta}^{\mathrm{T}} \boldsymbol{\lambda} \;, \end{aligned}$$

where tr$(\cdot)$ is the trace of a matrix and I is the identity matrix. Overall, the objective function for the k-th label becomes

```
Input:
  $\hat{Y}$ : $n \times m$ partial label matrix
  $W$: $n \times n$ PSD similarity matrix
Output:
  $F$: $n \times m$ predicted label matrix
Process:
  Decompose $W$ as $W = \sum_i \eta_i \boldsymbol{v}_i \boldsymbol{v}_i^{\mathrm{T}}$.
  $C = \sum_i \eta_i$, $W_i = \boldsymbol{v}_i \boldsymbol{v}_i^{\mathrm{T}}$.
  for $1 \leq k \leq m$
    $u_i = \hat{Y}_{\cdot k}^{\mathrm{T}} \boldsymbol{v}_i \boldsymbol{v}_i^{\mathrm{T}} \hat{Y}_{\cdot k}$.
    Solve
        min   $\gamma \boldsymbol{\lambda}^{\mathrm{T}} I \boldsymbol{\lambda} - \big(\boldsymbol{u} + 2\gamma\boldsymbol{\eta}\big)^{\mathrm{T}} \boldsymbol{\lambda}$
        s.t.  $\mathbf{1}^{\mathrm{T}} \boldsymbol{\lambda} = C, \quad \lambda_i \geq 0$
    $W^k = \sum_i \lambda_i W_i$, $L^k = D^k - W^k$
    Solve
        min   $\boldsymbol{f}^{\mathrm{T}} \big(\alpha L^k + \beta \Upsilon_k\big) \boldsymbol{f} - 2\big(\beta \hat{Y}_{\cdot k} + \mathbf{1}\big)^{\mathrm{T}} \boldsymbol{f}$
        s.t.  $\boldsymbol{f} \in [0,1]^n$
    Vary the threshold and choose the discretized $\boldsymbol{f}$ minimizing the above object function as $F_{\cdot k}$.
  end for
```

Figure 1: Pseudo-code of the WELL method

[1] For simplicity of discussion, we drop the superscript k for $\boldsymbol{\lambda}^k$.

[2] In the derivation we omit the terms being constant with $\boldsymbol{\lambda}$.

Table 1: Experimental results (mean±std) on Yeast data. ↑ indicates "the larger, the better"; ↓ indicates "the smaller, the better". The best performance and its comparable performances are bolded (statistical significance examined via pairwise t-tests at 95% significance level).

	WL	Approaches						
	Ratio	Well	WellMinus	ML-kNN	RankSVM	Elkan08	iter-SVM	bias-SVM
Hamming Loss$^{\downarrow}$	20%	**0.197±0.001**	0.203±0.008	0.297±0.002	0.207±0.001	0.270±0.025	0.208±0.000	0.208±0.000
	30%	**0.162±0.000**	0.167±0.005	0.275±0.005	0.169±0.000	0.239±0.131	0.170±0.000	0.170±0.000
	40%	**0.139±0.001**	0.144±0.005	0.260±0.010	0.150±0.001	0.237±0.039	0.148±0.000	0.148±0.000
	50%	**0.083±0.000**	**0.084±0.002**	0.210±0.005	0.088±0.002	0.188±0.043	0.085±0.000	0.086±0.000
	60%	**0.074±0.000**	**0.075±0.002**	0.198±0.001	0.080±0.001	0.154±0.001	0.077±0.000	0.077±0.000
Macro-F1$^{\uparrow}$	20%	**0.523±0.002**	0.506±0.021	0.033±0.007	0.498±0.003	0.515±0.001	0.491±0.000	0.491±0.000
	30%	**0.644±0.001**	0.632±0.012	0.153±0.022	0.627±0.001	0.619±0.122	0.623±0.000	0.623±0.000
	40%	**0.703±0.003**	0.690±0.012	0.249±0.062	0.678±0.003	0.591±0.032	0.681±0.000	0.681±0.000
	50%	**0.862±0.000**	0.859±0.005	0.480±0.023	0.849±0.004	0.702±0.053	0.856±0.000	0.855±0.000
	60%	**0.877±0.000**	0.874±0.005	0.536±0.011	0.866±0.002	0.754±0.001	0.872±0.000	0.871±0.000
Micro-F1$^{\uparrow}$	20%	**0.517±0.002**	0.495±0.029	0.037±0.011	0.481±0.003	0.507±0.002	0.475±0.000	0.475±0.000
	30%	**0.636±0.000**	0.619±0.017	0.179±0.029	0.612±0.001	0.626±0.123	0.608±0.000	0.608±0.000
	40%	**0.706±0.002**	0.687±0.015	0.277±0.065	0.672±0.003	0.612±0.031	0.676±0.000	0.676±0.000
	50%	**0.842±0.000**	0.838±0.006	0.520±0.019	0.830±0.003	0.706±0.057	0.835±0.000	0.834±0.000
	60%	**0.861±0.000**	0.857±0.005	0.576±0.009	0.849±0.002	0.767±0.001	0.854±0.000	0.854±0.000

$$\begin{aligned} \min_{\boldsymbol{\lambda}} \quad & \gamma \boldsymbol{\lambda}^{\mathrm{T}} I \boldsymbol{\lambda} - \big(\boldsymbol{u} + 2\gamma\boldsymbol{\eta}\big)^{\mathrm{T}} \boldsymbol{\lambda} \\ \text{s.t.} \quad & \mathbf{1}^{\mathrm{T}}\boldsymbol{\lambda} = C, \quad \lambda_i \geq 0 \, . \end{aligned} \tag{5}$$

Note that Eq. 5 has only equation constraint and I is very sparse. So, this QP problem can be solved very efficiently by SMO (sequential minimal optimization) (Platt 1999). The pseudo-code of the Well is summarized in Figure 1.

Empirical Study

We compare the Well method with state-of-the-art multi-label learning methods RankSVM (Elisseeff and Weston 2002) and ML-kNN (Zhang and Zhou 2007), and PU-learning methods Elkan08 (abbreviated for the method in (Elkan and Noto 2008)), iter-SVM and bias-SVM (Liu et al. 2003). We also evaluate a degenerated version of Well, denoted as WellMinus, where the original similarity matrix W is shared for all the labels, to study the utility of exploiting label correlation.

We use multi-label classification criteria *Hamming Loss*, *Macro-F1* and *Micro-F1* to measure the performance. Hamming Loss evaluates how many times an instance-label pair is misclassified; Macro-F1 averages the F1 measure on the predictions of different labels; Micro-F1 calculates the F1 measure on the predictions of different labels as a whole. Details can be found in (Zhang and Zhou 2010). Three real-world tasks, i.e., gene functional analysis, text classification and image annotation, are included in experiments. On each data set, we vary the *weak label ratio* (WL ratio), defined as $\|\hat{Y}_{i\cdot}\|_1/\|Y_{i\cdot}\|_1$, of each instance from 20% to 60% with 10% as interval to study the performance of different methods.

For Well, α and β are fixed as 100 and 10, respectively. This setting is sub-optimal and we will study how to set the parameters better in the future. We observed that the performance does not change much as α and β vary around the fixed values. Another parameter, γ, is tuned from $\{10^i | i = 0, 1, \cdots, 4\}$ based on the best performance on kernel alignment using five-fold cross-validation on training data [3]. The original similarity matrix used in Well and the kernel used in SVM-based methods are rbf kernels and the kernel width is fixed to 1. The SVM with rbf kernel used in all SVM-based methods is implemented by libSVM (Lin, Lin, and Weng 2007). Parameters of WellMinus are set to the same values as those for Well. For ML-kNN, we set $k = 10$ as suggested in (Zhang and Zhou 2007). For other parameters of the compared methods, we choose from the pool of $\{10^i | i = -4, -3, \cdots, 3, 4\}$ according to the best performance on Hamming Loss on the ground-truth. Note that by such a parameter setting, the comparison is unfavorable to our Well method; however, we will see that even in such setting the performance of Well is still superior to the compared methods.

Yeast Gene Functional Analysis

The first task is to predict the gene functional classes of the Yeast *Saccharomyces cerevisiae*. The Yeast data set investigated in (Elisseeff and Weston 2002) is used. The data set we used here contains 1,500 examples and 14 class labels. The average number of labels for each instance is 4.24±1.57.

Results are shown in Table 1. It can be seen that Well performs significantly better than all the other approaches except WellMinus on Hamming Loss when the WL ratio is larger than 50%. As the WL ratio decreases, the advantage of Well to other methods becomes more apparent. One reason is that Well uses not only the similarity between instances but also the similarity between labels.

Text Classification

The second task is a text classification task in SIAM Text Mining Competition (TMC) 2007. This data set is a subset of the Aviation Safety Reporting System (ASRS) data

[3] Hamming Loss or F1 could not be used to tune the parameter since there is no negative labeled data; while the unlabeled data will not affect the kernel alignment as we have demonstrated before.

Table 2: Experimental results (mean±std) on TMC data. ↑ indicates "the larger, the better"; ↓ indicates "the smaller, the better". The best performance and its comparable performances are bolded (statistical significance examined via pairwise t-tests at 95% significance level).

	WL	Approaches						
	Ratio	WELL	WELLMINUS	ML-kNN	RankSVM	Elkan08	iter-SVM	bias-SVM
Hamming Loss$^{\downarrow}$	20%	**0.163±0.002**	0.165±0.000	0.165±0.000	0.165±0.000	0.617±0.031	0.165±0.000	0.165±0.000
	30%	**0.130±0.001**	0.141±0.000	0.141±0.000	0.141±0.000	0.609±0.098	0.141±0.000	0.141±0.000
	40%	**0.091±0.001**	0.098±0.000	0.098±0.000	0.098±0.000	0.454±0.025	0.098±0.000	0.098±0.000
	50%	**0.072±0.000**	0.073±0.000	0.073±0.000	0.073±0.000	0.419±0.066	0.073±0.000	0.073±0.000
	60%	**0.067±0.000**	0.068±0.000	0.068±0.000	0.068±0.000	0.520±0.024	0.068±0.000	0.068±0.000
Macro-F1$^{\uparrow}$	20%	**0.479±0.012**	0.471±0.000	0.471±0.000	0.471±0.000	0.357±0.005	0.471±0.000	0.471±0.000
	30%	**0.622±0.011**	0.565±0.000	0.565±0.000	0.565±0.000	0.331±0.047	0.565±0.000	0.565±0.000
	40%	**0.782±0.002**	0.747±0.000	0.747±0.000	0.747±0.000	0.388±0.023	0.747±0.000	0.747±0.000
	50%	**0.821±0.000**	0.816±0.000	0.816±0.000	0.816±0.000	0.405±0.028	0.816±0.000	0.816±0.000
	60%	**0.832±0.001**	0.827±0.000	0.827±0.000	0.827±0.000	0.352±0.013	0.827±0.000	0.827±0.000
Macro-F1$^{\uparrow}$	20%	**0.481±0.014**	0.471±0.000	0.471±0.000	0.471±0.000	0.362±0.006	0.471±0.000	0.471±0.000
	30%	**0.641±0.014**	0.579±0.000	0.579±0.000	0.579±0.000	0.336±0.048	0.579±0.000	0.579±0.000
	40%	**0.783±0.003**	0.741±0.000	0.741±0.000	0.741±0.000	0.399±0.022	0.741±0.000	0.741±0.000
	50%	**0.824±0.000**	0.817±0.000	0.817±0.000	0.817±0.000	0.420±0.033	0.817±0.000	0.817±0.000
	60%	**0.839±0.001**	0.834±0.000	0.834±0.000	0.834±0.000	0.363±0.010	0.834±0.000	0.834±0.000

set, which contains a huge number of documents [4]. Each document is an aviation safety report documenting one or more problems that occurred on certain flights. The goal is to label the documents with respect to what types of problems they describe. Each document may belong to more than one class. Here we use the pre-processed version [5]. The data set we used here contains 1,000 examples and 15 class labels. The average number of labels for each instance is 3.57±0.73. The dimensionality of this data set is high (30,438), and therefore we first perform PCA to reduce the dimensionality to 7,000.

Results are summarized in Table 2. It can be seen that WELL performs significantly better than all other methods on all the evaluation criteria.

Image Annotation

The third task is image annotation. The data set we used was released by Microsoft Research Asia (MSRA). Each image is described by seven descriptors (i.e., 7 groups of features) including color moment, correlogram, EDH, face, HSV, RGB and wavelet texture. The total number of features is 899. All the labels are annotated by human. The data set we used here contains 1,000 examples and 15 class labels. The average number of labels for each instance in this subset is 6.760±0.94.

Results are summarized in Table 3. It can be seen that WELL always performs the best on all the criteria. Figure 2 shows two examples of the results. The PU-learning methods, Elkan08 and bias-SVM, predict much more labels than the ground-truth. It is probably because those methods are designed to handle balanced data which is unusual in the case of multi-label learning. Also, note that although bias-SVM and Elkan08 predict many labels, bias-SVM misses *clothing* for the first image and Elkan08 misses *leaf* for the second image. *clothing* is related to *women* and *leaf* is related to *jungle*; this implies that those methods have not utilized the label correlation well. The outputs of iter-SVM are as same as the inputs. This may be caused by that it treats the unlabeled data as negative and uses a self-training strategy, while the empirical loss is small at the beginning and thus the outputs will not change greatly during the iterations. Methods designed for classic multi-label learning, ML-kNN and RankSVM, predict almost as same as the inputs. This is not difficult to understand since they are designed for full label setting and tend to regard unassigned labels as negative. WELLMINUS also outputs more labels than the ground-truth, which may be caused by that it assumes all labels share the same similarity matrix and so the boundaries corresponding to different labels may be similar.

Conclusion

In this paper, we study the weak label problem which is a new kind of multi-label learning problem, where only a partial label set associated with each training example is provided. We propose the WELL method which considers the inherent class imbalance of the weak label problem and enforces the classification boundary for each label to go across low density regions. We formulate the objective as a quadratic programming problem which can be solved efficiently. To exploit label correlations, we assume that there is a group of low-rank base similarities, and the appropriate similarities between instances for different labels can be derived from these base similarities. Improving the efficiency of our method and applying it to large scale database are interesting issues for future study.

References

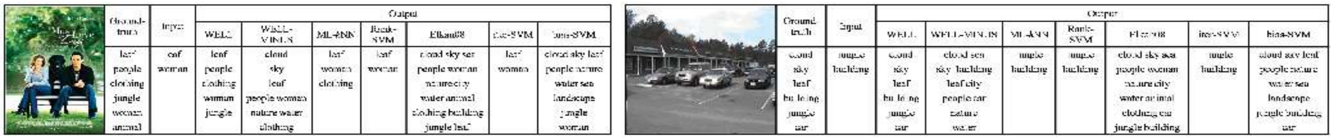
Belkin, M.; Niyogi, P.; and Sindhwani, V. 2006. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research* 7:2399–2434.

[4] `http://www.cs.utk.edu/tmw07/`

[5] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/multilabel.html`

Table 3: Experimental results (mean±std) on MSRA data. ↑ indicates "the larger, the better"; ↓ indicates "the smaller, the better". The best performance and its comparable performances are bolded (statistical significance examined via pairwise t-tests at 95% significance level).

	WL Ratio	Approaches						
		WELL	WELLMINUS	ML-kNN	RankSVM	Elkan08	iter-SVM	bias-SVM
Hamming Loss↓	20%	**0.249±0.004**	0.315±0.000	0.434±0.001	0.316±0.000	0.549±0.000	0.318±0.000	0.316±0.001
	30%	**0.224±0.007**	0.261±0.002	0.419±0.003	0.277±0.000	0.499±0.072	0.278±0.000	0.277±0.000
	40%	**0.198±0.007**	0.212±0.004	0.380±0.004	0.237±0.000	0.515±0.028	0.238±0.000	0.237±0.000
	50%	**0.155±0.001**	0.155±0.000	0.312±0.013	0.174±0.000	0.371±0.004	0.175±0.000	0.174±0.000
	60%	**0.127±0.003**	0.128±0.001	0.265±0.001	0.142±0.000	0.284±0.078	0.142±0.000	0.142±0.000
Macro-F1↑	20%	**0.634±0.009**	0.473±0.001	0.068±0.002	0.464±0.000	0.619±0.000	0.460±0.000	0.585±0.171
	30%	**0.676±0.015**	0.591±0.003	0.129±0.016	0.553±0.000	0.588±0.044	0.550±0.001	0.553±0.000
	40%	**0.726±0.012**	0.695±0.006	0.276±0.010	0.644±0.000	0.554±0.036	0.642±0.000	0.644±0.000
	50%	**0.800±0.002**	0.799±0.000	0.488±0.033	0.761±0.000	0.645±0.008	0.760±0.000	0.762±0.000
	60%	**0.842±0.003**	0.841±0.001	0.606±0.002	0.814±0.000	0.729±0.049	0.814±0.000	0.814±0.000
Micro-F1↑	20%	**0.643±0.009**	0.472±0.001	0.075±0.003	0.460±0.000	0.621±0.000	0.455±0.000	0.584±0.176
	30%	**0.688±0.014**	0.599±0.002	0.141±0.016	0.556±0.000	0.593±0.040	0.553±0.000	0.556±0.000
	40%	**0.732±0.012**	0.699±0.006	0.292±0.011	0.643±0.000	0.560±0.035	0.641±0.000	0.643±0.000
	50%	**0.802±0.002**	0.801±0.000	0.504±0.033	0.760±0.000	0.650±0.007	0.759±0.000	0.760±0.000
	60%	**0.843±0.003**	0.842±0.001	0.618±0.002	0.813±0.000	0.732±0.048	0.813±0.000	0.814±0.000



Ground-truth	Input	Output						
		WELL	WELL-MINUS	ML-kNN	Rank-SVM	Elkan08	iter-SVM	bias-SVM
leaf people clothing jungle woman animal	leaf woman	leaf people clothing woman jungle	cloud sky leaf people woman nature water clothing	leaf woman clothing	leaf woman	cloud sky sea people woman nature city water animal clothing building jungle leaf	leaf woman	cloud sky leaf people nature water sea landscape jungle woman

Ground-truth	Input	Output						
		WELL	WELL-MINUS	ML-kNN	Rank-SVM	Elkan08	iter-SVM	bias-SVM
cloud sky leaf building jungle car	jungle building	cloud sky leaf building jungle car	cloud sea sky building leaf city people car nature water	jungle building	jungle building	cloud sky sea people woman nature city water animal clothing car jungle building	jungle building	cloud sky leaf people nature water sea landscape jungle building car

Figure 2: Examples from MSRA data set. The ground-truth, input and predicted labels are shown on the right side of each image.

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

Elisseeff, A., and Weston, J. 2002. A kernel method for multi-labelled classification. In *NIPS 14*. 681–687.

Elkan, C., and Noto, K. 2008. Learning classifiers from only positive and unlabeled data. In *KDD*, 213–220.

Fung, G. P. C.; Yu, J. X.; Lu, H.; and Yu, P. S. 2006. Text classification without negative examples revisit. *IEEE Trans Knowledge and Data Engineering* 18(1):6–20.

Ghamrawi, N., and McCallum, A. 2005. Collective multi-label classification. In *CIKM*, 195–200.

Ji, S.; Tang, L.; Yu, S.; and Ye, J. 2008. Extracting shared subspace for multi-label classification. In *KDD*, 381–389.

Joachims, T. 1998. Text categorization with support vector machines: learning with many relevant features. In *ECML*, 137–142.

Lee, W. S., and Liu, B. 2003. Learning with positive and unlabeled examples using weighted logistics regression. In *ICML*, 448–455.

Li, X., and Liu, B. 2003. Learning to classify texts using positive and unlabeled data. In *IJCAI*, 587–594.

Lin, H.-T.; Lin, C.-J.; and Weng, R. C. 2007. A note on Platt's probabilistic outputs for support vector machines. *Machine Learning* 68(3):267–276.

Liu, B.; Dai, Y.; Li, X.; Lee, W. S.; and Yu, P. S. 2003. Building text classifiers using positive and unlabeled examples. In *ICDM*, 19–22.

Liu, J.; Chen, J.; Chen, S.; and Ye, J. 2009. Learning the optimal neighborhood kernel for classification. In *IJCAI*, 1144–1149.

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

Platt, J. C. 1999. Fast training of support vector machines using sequential minimal optimization. In Schölkopf, B.; Burges, C. J. C.; and Smola, A. J., eds., *Advances in Kernel Methods*. MIT Press. 185–208.

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

Shi, J., and Malik, J. 2000. Normalized cuts and image segmentation. *IEEE Trans Pattern Analysis and Machine Intelligence* 22(8):888–905.

Sun, L.; Ji, S.; and Ye, J. 2008. Hypergraph spectral learning for multi-label classification. In *KDD*, 668–676.

Ueda, N., and Saito, K. 2003. Parametric mixture models for multi-labeled text. In *NIPS 15*. 721–728.

Yan, R.; Těsić, J.; and Smith, J. R. 2007. Model-shared subspace boosting for multi-label classification. In *KDD*, 834–843.

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

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, Y., and Zhou, Z.-H. 2010. Multi-label dimensionality reduction via dependence maximization. *ACM Trans Knowledge Discovery from Data*.

Zhou, Z.-H.; Zhang, M.-L.; Huang, S.-J.; and Li, Y.-F. 2008. MIML: A framework for learning with ambiguous objects. CORR abs/0808.3231.

Zhu, S.; Ji, X.; Xu, W.; and Gong, Y. 2005. Multi-labelled classification using maximum entropy method. In *SIGIR*, 274–281.