Robust Multi-Label Learning with PRO Loss

Miao Xu, Yu-Feng Li, and Zhi-Hua Zhou, *Fellow, IEEE*

Abstract—Multi-label learning methods assign multiple labels to one object. In practice, in addition to differentiating relevant labels from irrelevant ones, it is often desired to rank *relevant* labels for an object, whereas the ranking of *irrelevant* labels is not important. Thus, we require an algorithm to do classification and ranking of relevant labels simultaneously. Such a requirement, however, cannot be met because most existing methods were designed to optimize existing criteria, yet there is no criterion which encodes the aforementioned requirement. In this paper, we present a new criterion, PRO LOSS, concerning the prediction of all labels as well as the ranking of only relevant labels. We then propose ProSVM which optimizes PRO LOSS efficiently using alternating direction method of multipliers. We further improve its efficiency with an upper approximation that reduces the number of constraints from $O(T^2)$ to $O(T)$, where T is the number of labels. We then notice that in real applications, it is difficult to get full supervised information for multi-label data. To make the proposed algorithm more robust to supervised information, we adapt ProSVM to deal with the multi-label learning with partial labels problem. Experiments show that our proposal is not only superior on PRO LOSS, but also highly competitive on existing evaluation criteria.

Index Terms—Multi-label learning, learning criterion, partial labels, PRO LOSS, ProSVM

1 INTRODUCTION

In real applications, one object may be associated with multiple labels simultaneously, and such problems are realized by multi-label learning [1]. During the past decade, many multi-label methods have been developed and found useful in diverse applications [2], [3], [4].

For a multi-label task, generally one object is associated with a subset of labels; we call these labels as *relevant* ones whereas the remaining as *irrelevant* ones. The basic goal of multi-label learning is usually label prediction, that is, to predict which labels are relevant and which are irrelevant. In many applications, however, in addition to label prediction, there is often another requirement, i.e., to get a good ranking of the predicted relevant labels. Consider a simple example in Fig. 1. Both images have the relevant labels *mountain*, *cattle* and *road*, whereas their focuses are quite different. To better describe these images, in addition to predicting which labels are relevant, it would be better to get their relevant labels' rankings as well, that is, {*cattle*, *mountain*, *road*} for the left one and {*mountain*, *road*, *cattle*} for the right one.

Although the ranking of relevant labels is important, correct ranking of all the labels, which is classically considered in label ranking problems [5], is not necessary in multi-label learning. The reason is that irrelevant labels do not occur within any image; thus their ranking will make no sense. Taking Fig. 1 again as an example, assume we have irrelevant labels *sea*, *ship* and *pyramid*. In this way, whether *ship* should be ranked before *sea* or *pyramid* is pointless. Thus although we need to consider the ranking of relevant labels, the ranking of irrelevant labels, which does not occur within any image, is not useful.

Regarding the ranking of relevant labels, we want to emphasize that existing works focusing on top-predicted labels [6], [7] could not be used to address this problem properly. Such kind of works emphasized on the ranking of top-k ranked labels, where k is a fixed number. In our requirement here, we need to adaptively decide which labels are relevant and focus on the ranking of *all relevant* labels, while the number of relevant labels may be larger or smaller than the simply fixed number k.

Besides existing works focusing on top-predicted labels, other existing methods cannot address such a learning problem either. They either focused on the label prediction, ignoring the ranking of relevant labels, or provided a ranking for all or a fixed number of labels, without differentiating relevant labels from irrelevant ones. Considering the ranking of all the labels also introduces overfitting and computational burden because the ranking of irrelevant labels is unnecessary. So how to design an algorithm to solve our concerned problem? We know that most algorithms are designed to optimize a specific learning criterion, and the infeasibility of existing methods on our concerned problem is owe to the fact that they were designed to optimize the classical performance criteria. For example, BR [8], [9] was tailored for HAMMING LOSS; GFM [10] was designed for F1; RankSVM [11] was designed for RANKING LOSS; EncDec [12] was designed to optimize Subset Accuracy. As we will discuss comprehensively in the next section, none of the classical criteria is able to express the requirement of our concerned problem precisely.

- *M. Xu is with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing Shi, Jiangsu Sheng 210023, China, the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210023, China, and also with the RIKEN Center for Advanced Intelligence Project, Tokyo 103-0027, Japan. E-mail: xum@lamda.nju.edu.cn.*
- *Y.-F. Li and Z.-H. Zhou are with the National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing Shi, Jiangsu Sheng 210008, China, and also with the Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing 210023, China. E-mail: {liyf, zhouzh}@lamda.nju.edu.cn.*

Manuscript received 21 Feb. 2018; revised 21 Dec. 2018; accepted 3 Mar. 2019. Date of publication 0 . 0000; date of current version 0 . 0000.
(Corresponding author: Zhi-Hua Zhou.)
Recommended for acceptance by J. Wen.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TKDE.2019.2908898

Fig. 1. Rankings of relevant labels in images. Left: {*cattle*, *mountain*, *road*}. Right: {*mountain*, *road*, *cattle*}.

Therefore, to address our problem, new criteria as well as new algorithms are needed.

Another problem with multi-label learning in practice is that it is hard to get the full annotation of multi-label data, especially when there are a large number of candidate labels [13]. Thus it is normal to have partial labels in multi-label learning. Giving the full annotation not only is expensive, but also requires labelers to be careful enough to annotate all candidate labels. Although there are some works focusing on the multi-label learning with partial labels problem [13], [14], [15], [16], [17], these methods focused on giving a good prediction while none of them could give both good classification and ranking of relevant labels. Thus in our problem under partial label scenarios, we need to provide ranking of relevant labels in addition to mere classification.

This paper presents the Prediction and Relevance Ordering Loss (PRO Loss), a new multi-label criterion that concerns the label prediction as well as the ranking of relevant labels. We then propose ProSVM, a large margin approach that employs alternating direction method of multipliers to optimize the PRO Loss efficiently. To further improve the efficiency, we introduce an upper approximation that reduces the number of constraints from $O(T^2)$ to $O(T)$ where T is the number of labels. To solve the partial label problem so as to make the proposal more robust to incomplete annotations, we extend the PRO Loss to partial label cases. We also propose optimization algorithms ProSVM-P to deal with the partial labels in training data under the *inductive setting* [18]. Experiments show that when we have perfect training data, our proposal is not only superior to state-of-the-art approaches on PRO Loss, but also highly competitive on existing evaluation criteria. We also extensively demonstrate the effectiveness of our proposed algorithms on various applications under partial label scenarios.

The rest of the paper is organized as follows. We will first introduce related work in Section 2. PRO Loss and ProSVMs are presented in Sections 3 and 4 respectively, followed by proposing ProSVM-P which can deal with partial labels in Section 5. Finally, we present the experimental results in Sections 6 and 7, followed by conclusion in Section 8.

Preliminary results of this paper have been reported in [19]. In this paper, our main contribution is that we have considered the relevance ordering problem with *partial labels*, which widely occurs in real applications and this line of study has not been presented before. We have also added corresponding optimization algorithms and experimental results. Besides these, we have further added the Critical Difference Diagram of our experimental results, illustrations of real images, rigorous proofs, additional empirical comparison with more recently proposed algorithms on larger data set, et al.

2 Related Work

2.1 Multi-Label Learning

Multi-label learning, which assumes one instance is associated with multiple labels, has diverse applications, e.g., text classification [2], genomics [3], image tagging [4], [20], action recognition [21], et al. For detailed survey of multi-label learning, please refer to [1].

The most straightforward solution to multi-label learning is the Binary Relevance (BR) method [8] which simply learned one binary classifier for each label. Although such a method is the most intuitive solution to multi-label learning [22], it has been criticized for ignoring the label dependence of multiple labels [1]. To take the label correlation into consideration, some works used label correlation directly from prior knowledge [23], or tried to identify them explicitly from data [24]. There are also a bunch of other important works considering the label correlation implicitly by learning multiple binary classifiers simultaneously in one framework and incorporating a regularization term into the optimization. One example is RankSVM [11], which used an SVM-style algorithm to optimize multiple classifiers for label pairs in one optimization. These algorithms have been shown effective in various applications.

Most of these multi-label learning algorithms were proposed to optimize existing multi-label learning criterion. For example, it was proved in [25] that the methods estimating the posterior distribution of single labels and multiple labels are tailored for Hamming Loss and Subset Accuracy respectively. In this way, BR [8], [9] optimized Hamming Loss. [12], [26] optimized Subset Accuracy. [27] and [28] optimized Ranking Loss. [10], [29] were designed for optimizing F1.

One straightforward solution to the problem of considering both prediction and ranking of relevant labels is to first employ a multi-label learning algorithm to do classification and then use some label ranking methods to rank the relevant labels. However, the objective of this paper is to propose the learning objective for such kind of problem, thus an optimization method can be proposed considering the ranking and classification problem in one framework. We will show in Section 6 that our one-framework optimization algorithm performs significantly better than the two-stage classification and ranking methods. The PC method [30] considered a combination of multi-label learning and label ranking by creating an additional calibrated label. However, it concerned either "multi-label learning" or "label ranking" without realizing that only the ranking of relevant labels is crucial. [31] proposed a related label ranking method GMLC which assumed that labels of all objects have fixed number of graded relevancies; in contrast, we do not assume the existence of such information.

2.2 Multi-Label Learning with Partial Labels

Many researchers these years have noticed that fully supervised information for multi-label learning is difficult to acquire. Multi-label learning with partial labels problem is a weakly supervised learning problem [32] when only a subset of all the labels are annotated, and different instances have different annotated subsets. In such a kind of learning problem, supervised information is not only *incomplete* but also *inexact* [32]. There are some works focusing directly on solving the multi-label learning with *partial labels* problem [20], [33], [34], [35], which has also been called learning with *incomplete label assignments* or *missing labels*. For example, [36] used the low-rank mapping to fulfill the partial labels; [33] proposed a disciplined approach to handle the partial labels; [37] required the recovered label space to be both low-rank and sparse; [16] used a label graph to propagate the known labels to the unknown labels; [17] combined the prediction using a principled way to make a safe use of unknown labels. However, all these works focus on solving the multi-label learning with partial labels problem and ignore the ranking of relevant labels. Moreover, most of them [14], [15], [35] worked under a *transductive setting* which involves the test data into the training process [18] while in this paper, we will work in an *inductive setting* where the test data can be seen only after the classifier has been learned, and we will solve the problem of acquiring a good classification and ranking of relevant labels simultaneously when there are only partial labels. We will show that our algorithm taking both classification and ranking into consideration gets a better performance compared to state-of-the-art multi-label learning methods designed for partial labels problem.

2.3 Existing Criteria

Suppose that we are given a set of n instances $\{\mathbf{x}_i\}_{i=1}^n$ and a set of T labels $L = \{l_1, \ldots, l_T\}$. Each instance $\mathbf{x}_i \in \mathbb{R}^d$ has the ranked relevant label set $R_i \subseteq L$ and corresponding irrelevant label set $\overline{R}_i = L - R_i$, on which ranking will not be concerned.

Existing multi-label learning algorithms typically learn a function $\mathbf{g}(\mathbf{x}_i) = [g_1(\mathbf{x}_i), \ldots, g_T(\mathbf{x}_i)]$ that will assign a real-valued score $g_t(\mathbf{x}_i)$ to each label $l_t, t \in \{1, \ldots, T\}$. The labels can then be ranked according to these scores. To further differentiate relevant labels from irrelevant ones, these algorithms need to additionally learn a threshold score, denoted by $g_\Theta(\mathbf{x}_i)$. Those labels with scores larger than the threshold will be regarded as relevant ones, or else irrelevant ones. Here $g_\Theta(\mathbf{x}_i)$ can be simply set to some fixed constant; or it can also be set more accurately by learning from data [30]. We denote all the predicted relevant labels as $\hat{R}_i$, i.e., $\hat{R}_i = \{l_t \in L | g_t(\mathbf{x}_i) > g_\Theta(\mathbf{x}_i)\}$.

In the following we will discuss existing multi-label criteria and their limitations regarding our concerned problem.

- Hamming Loss [9], [38]

$$\frac{1}{nT}\sum_{i=1}^{n} |\hat{R}_i \triangle R_i|.$$

Here $\triangle$ stands for the symmetric difference between two label subsets. Obviously, the Hamming Loss ignores the fact that relevant and irrelevant labels may have different priorities and relevant labels should be ranked.

- Ranking Loss [27], [28]

$$\frac{1}{n}\sum_{i=1}^{n} \frac{\sum_{(l_t,l_s)\in R_i\times\overline{R}_i} \delta[g_t(\mathbf{x}_i) < g_s(\mathbf{x}_i)]}{|R_i| \times |\overline{R}_i|}.$$

Here δ is the indicator function. Ranking Loss concerns the relative ranking of each relevant-irrelevant label pair. However, it does not consider the ranking of relevant labels.

- One-error [11], [39], [40]

$$\frac{1}{n}\sum_{i=1}^{n} \delta[l_{\arg\max_t g_t(\mathbf{x}_i)} \notin R_i].$$

One-error considers the top-predicted relevant label only and neglects all the other relevant labels. It can also be described as Top-1 Precision [6], [7].

- Average Precision [11], [39], [40]

$$\frac{1}{n}\sum_{i=1}^{n} \frac{1}{|R_i|} \sum_{t:l_t\in R_i} \frac{|\{l_s \in R_i | g_s(\mathbf{x}_i) > g_t(\mathbf{x}_i)\}|}{|\{l_s | g_s(\mathbf{x}_i) > g_t(\mathbf{x}_i)\}|}.$$

Average Precision does not concern the misclassification of relevant labels and irrelevant labels.

- Coverage [39]

$$\frac{1}{n}\sum_{i=1}^{n} \max_{t:l_t\in R_i} |\{s | g_s(\mathbf{x}_i) > g_t(\mathbf{x}_i)\}|.$$

Coverage concerns the position of the relevant label with lowest predicted score only, thus neglecting all the other relevant labels.

- Subset Accuracy [12], [25], [26]

$$\frac{1}{n}\sum_{i=1}^{n} \delta[\hat{R}_i = R_i].$$

Subset Accuracy does not consider label ranking.

- F1 [10], [29]

$$\frac{1}{n}\sum_{i=1}^{n} \frac{2|R_i \cap \hat{R}_i|}{|R_i| + |\hat{R}_i|}.$$

F1 does not take any ranking of relevant labels into account.

There are also some work focusing on the cost-sensitive multi-label learning, designing an algorithm which can adapt to different criteria easily [41]. However, current methods can only deal with some special performance measures, and do not consider the relevant labels' ranking information. There is another popular ranking evaluation criterion for multi-label learning, nDCG@k used in [6]. nDCG@k is a popular performance measure for extreme multi-label learning and a lot of algorithms have been reported to perform good on this measurement [6], [42], [43], [44]. Although nDCG@k is also able to evaluate the ranking of relevant labels, the difference between nDCG@k and our required loss lies in the setup of k. In nDCG@k, k is often known in advance and remains the same across all instances. In our requirement, the number of relevant labels should be adaptively determined for different instances, instead of being a simply fixed integer.

It is evident that all the above criteria fail to express our requirement, i.e., attaining an accurate label prediction and a correct ranking of relevant labels without being affected by the ranking of irrelevant labels. To the best of our knowledge, this is the first study on this problem.

3 PRO Loss

We first introduce some notations. Given an instance $\mathbf{x}$ and its relevant label set R, to characterize the ranking on R, we

TABLE 1
Examples Showing the Effects of Different Multi-Label Criteria on the Left Image in Fig. 1

*	Prediction	Outputs l_1	l_2	l_3	l_4	l_5	Θ	Pro	Hamm.	Rank.	OneE.	AveP.	Covr.	SubA.	F1
1	*cattle > mountain > road*	5	4	3	2	1	2.5	0.000	0.000	0.000	0.000	1.000	2.000	1.000	1.000
2	*cattle > road > mountain*	5	3	4	2	1	2.5	0.083	0.000	0.000	0.000	1.000	2.000	1.000	1.000
3	*cattle > mountain*	5	4	3	2	1	3.5	0.083	0.200	0.000	0.000	1.000	2.000	0.000	0.800
4	*cattle > mountain > road > car*	5	4	3	2	1	1.5	0.125	0.200	0.000	0.000	1.000	2.000	0.000	0.857
5	*sea > car*	1	2	3	4	5	3.5	1.000	1.000	1.000	1.000	0.478	4.000	0.000	0.000

There are five candidate labels, in which l_1 =cattle, l_2 =mountain, and l_3 =road are relevant labels ranked as cattle > mountain > road, and l_4 =car and l_5 =sea are irrelevant labels. Outputs are the scores of each label. The larger the score, the higher the label ranked. Θ is the threshold to differentiate relevant labels from irrelevant ones. Hamm., Rank., OneE., AveP., Covr., and SubA. are abbreviations for Hamming Loss, Ranking Loss, One Error, Average Precision, Coverage, and Subset Accuracy respectively.

denote by $\prec_{\mathbf{x}}(a)$ the set of indices of labels that are less relevant than l_a. The threshold, denoted as Θ whose predicted value $g_\Theta(\mathbf{x})$ should be larger than the scores of irrelevant labels and smaller than those of relevant labels, can be seen as a pseudo label, which should be *more relevant* than irrelevant labels and *less relevant* than relevant labels. Specially, suppose l_1 and l_2 are relevant labels and l_1 is more relevant than l_2, while l_3 and l_4 are the irrelevant labels, we have $\prec_{\mathbf{x}}(1)=\{2,\Theta,3,4\}$, $\prec_{\mathbf{x}}(2)=\{\Theta,3,4\}$, $\prec_{\mathbf{x}}(\Theta)=\{3,4\}$ and $\prec_{\mathbf{x}}(3)=\prec_{\mathbf{x}}(4)=\emptyset$.

In multi-label learning for an instance $\mathbf{x}$, label l_a (where a is the index of label) can be either relevant, irrelevant, or the pseudo label used to differentiate relevant labels from irrelevant ones. Therefore one can divide all the labels into three subgroups, that is, relevant labels, irrelevant labels, and pseudo label. $\mathcal{B}_{\mathbf{x}}(a)$ maps a label l_a to one of the three subgroups (relevant, irrelevant or pseudo label). Back to the $\{l_1, l_2, l_3, l_4\}$ example mentioned in the above paragraph, we can have $\mathcal{B}_{\mathbf{x}}(1)=\mathcal{B}_{\mathbf{x}}(2)=\{1,2\}$, $\mathcal{B}_{\mathbf{x}}(3)=\mathcal{B}_{\mathbf{x}}(4)=\{3,4\}$ and $\mathcal{B}_{\mathbf{x}}(\Theta)=\{\Theta\}$. We then define the PRO Loss for an instance $\mathbf{x}$ as

$$\mathcal{L}(R,\prec,\mathbf{g})=\sum_{l_t\in R\cup\{\Theta\}}\sum_{s\in\prec_{\mathbf{x}}(t)}\frac{1+\delta[\mathcal{B}_{\mathbf{x}}(t)=\mathcal{B}_{\mathbf{x}}(s)]}{4|\mathcal{B}_{\mathbf{x}}(t)|\times|\mathcal{B}_{\mathbf{x}}(s)-\{t\}|}\ell_{t,s}. \tag{1}$$

Here $\ell_{t,s}$ refers to a modified 0-1 error. Specifically, $\ell_{t,s}=1$ if $g_t(\mathbf{x})<g_s(\mathbf{x})$, $\frac{1}{2}$ if $g_t(\mathbf{x})=g_s(\mathbf{x})$[1] and 0 otherwise. Essentially, PRO Loss is the weighted counting of incorrectly ranked label pairs.

As can be seen, besides the relevant-irrelevant label pairs considered in Ranking Loss and the label-threshold pairs considered in Hamming Loss, PRO Loss further considers the relevant-relevant label pairs. It is noteworthy that the ranking of irrelevant labels is not valued in Eq. (1). Hence, PRO Loss considers an accurate label prediction as well as a correct ranking of only relevant labels.

To balance these label pairs to avoid the situation that one term dominates all others, we normalize four types of label pairs, i.e., (*relevant*, *relevant*), (*relevant*, *irrelevant*), (*relevant*, *threshold*) and (*threshold*, *irrelevant*), by their respective set sizes. Note that the set sizes of these four label pairs are $|R|(|R|-1)/2$, $|R||\overline{R}|$, $|R|$ and $|\overline{R}|$, respectively, which can be written in a general form as

1. When $g_t(\mathbf{x})=g_s(\mathbf{x})$, neither "$l_t$ is more relevant than l_s" nor "l_s is more relevant than l_t" is judged; thus, we assign the error as $1/2$ on average.

$$h_{t,s}=\frac{|\mathcal{B}_{\mathbf{x}}(t)|\times|\mathcal{B}_{\mathbf{x}}(s)-\{t\}|}{1+\delta[\mathcal{B}_{\mathbf{x}}(t)=\mathcal{B}_{\mathbf{x}}(s)]}.$$

To further normalize the sum of these weighted pairs' losses to be within the range of [0,1], we divide the weighted sum by a factor of 4 which equals the number of types of different label pairs. This leads to our PRO Loss.

We will use some examples in Table 1 to illustrate the merit of PRO Loss compared to existing multi-label criteria. The example used is the left image of Fig. 1. There are 5 candidate labels, in which l_1 =*cattle*, l_2 =*mountain* and l_3 =*road* are relevant labels ranked as *cattle* > *mountain* > *road*, and l_4 =*car* and l_5 =*sea* are irrelevant labels. Outputs are the scores of each label. The larger the score, the higher the label ranked. Θ is the threshold to differentiate relevant labels from irrelevant ones.

The Output 1 in Table 1 gives the output perfectly agreeing with the ground truth. We can see that all the criteria give the best evaluation, showing the effectiveness of all these criteria for the perfect output. However, when we have a look at Output 2, where the ranking of relevant labels is *incorrect* while the classification is correct, we can find that only PRO Loss punishes such kind of error while all other criteria give an evaluation having no difference from that of the perfect Output 1. We can conclude that existing multi-label criteria cannot penalize the wrongly ranked relevant labels, while PRO Loss can.

In Output 3, one relevant label is classified as irrelevant, but the ranking of all labels remains unchanged compared to Output 1 according to their predicted scores. In this way, PRO Loss, Hamming Loss and F1 penalize such kind of error, while other criteria still give the "perfect" evaluation. This phenomenon tells us that Ranking Loss, One-Error, Average Precision and Coverage only care about the ranking of labels, while nothing is paid when the classification is wrong. PRO Loss, Hamming Loss, and F1, on the contrary, penalize the classification error.

In Output 3 and Output 4, we can see that misclassification happens on relevant label and irrelevant label respectively, resulting in same Hamming Loss, but different PRO Loss and F1. In this example, even though the number of relevant labels is similar to that of irrelevant labels, we can have different F1 and PRO Loss which measure different types of classification errors (i.e., relevant or irrelevant) while Hamming Loss remains unchanged. For some real multi-label datasets, the number of relevant labels may be much smaller than that of irrelevant labels. In this way, treating the relevant and irrelevant labels equally (as HAMMING LOSS does) will make the misclassification of relevant labels insignificant. We want to mention that although F1 have the advantage of treating relevant and irrelevant labels unequally just as PRO LOSS, it cannot penalize the wrongly ranked relevant labels, as we have shown in Output 2.

Comparing Output 4 and Output 5, although Output 4 only misclassifies one label, its SUBSET ACCURACY is equal to that of Output 5 in which none of the labels is correctly classified. We can easily see that SUBSET ACCURACY is too strict to reward the almost-correct output of multi-label learning algorithms.

In a word, PRO LOSS concerns both classification of all the labels and ranking of relevant labels as we have shown in the examples in Table 1, while none of the other existing multi-label criteria can fulfill the requirement compared to PRO LOSS.

4 PROSVMS

Note that $\ell_{t,s}$, a modified 0-1 loss, is non-convex and difficult to optimize. Instead of optimizing the difficult non-convex PRO LOSS directly, we consider optimizing a large margin surrogate convex loss as follows:

$$\min_{\mathbf{g}} \quad \lambda \sum\nolimits_{i=1}^{n} \widehat{\mathcal{L}}(\mathbf{x}_i, R_i, \prec, \mathbf{g}) + \mathrm{Regularizer}(\mathbf{g}), \tag{2}$$

where $\mathrm{Regularizer}(\mathbf{g})$ is a regularizer for $\mathbf{g}$, $\widehat{\mathcal{L}}(\mathbf{x}_i, R_i, \prec, \mathbf{g}) = \sum_{l_t \in R_i \cup \{\Theta\}} \sum_{s \in \prec_{\mathbf{x}_i}(t)} (1 + g_s(\mathbf{x}_i) - g_t(\mathbf{x}_i))_+ / (4h_{s,t})$ is the surrogate convex loss of PRO LOSS, $(u)_+ = \max\{0, u\}$, and λ is a parameter trading off the functional complexity of $\mathbf{g}$ and the surrogate convex loss.

Without loss of generality, suppose g's are linear models, i.e., $g_t(\mathbf{x}) = \mathbf{w}_t^\top \mathbf{x}, t \in \{1, \ldots, T\} \cup \{\Theta\}$ and $\mathrm{Regularizer}(\mathbf{g}) = \sum_{t \in \{1,\ldots,T\} \cup \{\Theta\}} \|\mathbf{w}_t\|^2/2$. Let $\mathbf{w} \triangleq [\mathbf{w}_1; \ldots; \mathbf{w}_T; \mathbf{w}_\Theta]$ and let D be the training set. Noting that $\widehat{\mathcal{L}}(\mathbf{x}_i, R_i, \prec, \mathbf{g})$ is no more than a sum of hinge losses, Eq. (2) can then be cast into an SVM-type problem in the following general form:

$$\begin{aligned} \min_{\mathbf{w}, \boldsymbol{\xi}} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + \lambda \mathbf{C}^\top \boldsymbol{\xi}, \\ \text{s.t.} \quad & A\mathbf{w} \geq \mathbf{1}_p - \boldsymbol{\xi}, \quad \boldsymbol{\xi} \geq \mathbf{0}_p, \end{aligned} \tag{3}$$

where $p = nT + \sum_{i=1}^{n} |R_i|(2T - |R_i| - 1)/2$ is the total number of constraints, and $\mathbf{1}_p(\mathbf{0}_p)$ is the $p \times 1$ all one (zero) vector. The entries in vector $\mathbf{C}$ correspond to the weights of hinge losses and the matrix $\mathbf{A}$ corresponds to the constraints across instances.

Note that in Eq. (3), ξ does not need to be optimized since it can be easily determined by $\mathbf{w}$, hence Eq. (3) can be reformulated into the following form without ξ, i.e.,

$$\min_{\mathbf{w}} F(\mathbf{w}, D) \triangleq \frac{1}{2}\|\mathbf{w}\|^2 + \lambda \mathbf{C}^\top (\mathbf{1}_p - \mathbf{A}\mathbf{w})_+. \tag{4}$$

4.1 An Efficient Algorithm

Eq. (4) is large scale. Specifically, although matrix $\mathbf{A}$ is sparse, it can still involves $O(dnT^2)$ non-zero entries which is beyond the memory capability of computers even for medium-sized datasets. To address Eq. (4) memory efficient, we in this section consider an Alternating Direction Method of Multipliers (ADMM) solution.

ADMM [45] is a simple and efficient approach for large scale optimization. Its basic idea is to take the *decomposition-coordinate* procedure so that the solution to subproblems can be coordinated to find the solution to the original problem. Since subproblems can usually be memory efficient, ADMM is capable of approximating the solution to large scale problems via addressing small subproblems sequentially. Moreover, ADMM is easy to be parallelized. Recently, ADMM is found effective in a number of machine learning problems [46], [47].

Following the ADMM procedure, we first decompose the training set D into Z disjoint subsets, i.e., $\{D^1, \ldots, D^Z\}$, and then rewrite Eq. (4) into the following equivalent form:

$$\begin{aligned} \min_{\bar{\mathbf{w}}^0, \bar{\mathbf{w}}^1, \ldots, \bar{\mathbf{w}}^Z} \quad & \sum\nolimits_{z=1}^{Z} F(\bar{\mathbf{w}}^z, D^z), \\ \text{s.t.} \bar{\mathbf{w}}^z = \bar{\mathbf{w}}^0, \quad & \forall\ z = 1, \ldots, Z. \end{aligned} \tag{5}$$

By introducing the surrogate augmented lagrangian function [48] into Eq. (5), we have

$$\begin{aligned} \mathbb{L}(\{\bar{\mathbf{w}}^0, \ldots, \bar{\mathbf{w}}^Z\}, \{\alpha^z\}_{z=1}^{Z}, \eta) = & \sum_{z=1}^{Z} F(\bar{\mathbf{w}}^z, D^z) \\ & + \sum_{z=1}^{Z} (\alpha^z)^\top (\bar{\mathbf{w}}^z - \bar{\mathbf{w}}^0) + \frac{\eta}{2} \sum_{z=1}^{Z} \|\bar{\mathbf{w}}^z - \bar{\mathbf{w}}^0\|^2, \end{aligned}$$

where α^z's and η are the lagrange multipliers. $\mathbb{L}$ is then solved in an alternative manner, i.e., updating the solutions to $\{\bar{\mathbf{w}}^1, \ldots, \bar{\mathbf{w}}^Z\}$, $\{\bar{\mathbf{w}}^0\}$ and $\{\alpha^z\}_{z=1}^{Z}$ separately and iteratively until the algorithm converges. Detailed processes are shown in Algorithm 1.

Algorithm 1. ProSVM

Decompose dataset D into Z disjoint subsets, i.e., $D^1, \ldots, D^Z$. Set $k = 0$.

Initialize $\{\bar{\mathbf{w}}_0^0, \bar{\mathbf{w}}_0^1, \ldots, \bar{\mathbf{w}}_0^Z, \alpha_0^1, \ldots, \alpha_0^Z\}$ as zeros.

while not converge **do**

Set $k = k + 1$ and update $\{\bar{\mathbf{w}}_k^0, \{\bar{\mathbf{w}}_k^z, \alpha_k^z\}_{z=1}^{Z}\}$ as:

$$\{\bar{\mathbf{w}}_k^z\}_{z=1}^{Z} = \operatorname*{arg\,min}_{\bar{\mathbf{w}}^1, \ldots, \bar{\mathbf{w}}^Z} \ \mathbb{L}(\bar{\mathbf{w}}_{k-1}^0, \{\bar{\mathbf{w}}^z, \alpha_{k-1}^z\}_{z=1}^{Z}, \eta) \tag{6}$$

$$\bar{\mathbf{w}}_k^0 = \operatorname*{arg\,min}_{\bar{\mathbf{w}}^0} \ \mathbb{L}(\bar{\mathbf{w}}^0, \{\bar{\mathbf{w}}_k^z, \alpha_{k-1}^z\}_{z=1}^{Z}, \eta) \tag{7}$$

$$\alpha_k^z = \alpha_{k-1}^z + \eta(\bar{\mathbf{w}}_k^z - \bar{\mathbf{w}}_k^0)^\top, \quad \forall z = 1, \ldots, Z$$

end while

Output the final $\bar{\mathbf{w}}^0$

In Algorithm 1 the key for us to design a competent ProSVM algorithm is to efficiently solve Eqs. (6) and (7). As for Eq. (6), it is equivalent to solving the following Z independent smaller subproblems

$$\min_{\bar{\mathbf{w}}^z} \quad F(\bar{\mathbf{w}}^z, D^z) + (\alpha_{k-1}^z)^\top \bar{\mathbf{w}}^z + \frac{\eta}{2}\|\bar{\mathbf{w}}^z - \bar{\mathbf{w}}_{k-1}^0\|^2, \tag{8}$$

which is a quadratic programming (QP) problem. Furthermore, noting that $\mathbf{A}$ is sparse and Eq. (8) is similar to standard SVM problem, Eq. (8) can be solved efficiently by state-of-art SVM solvers. As for Eq. (7), it has a closed-form solution, i.e.,

$\bar{\mathbf{w}}_k^0 = \sum_{z=1}^{Z}(\alpha_{k-1}^z + \eta\bar{\mathbf{w}}_k^z)/(\eta Z)$. Therefore, both Eqs. (6) and (7) can be solved efficiently.

4.2 Reducing the Number of Constraints

There are $O(T|R|)$ constraints in total for each instance in Eq. (2). Thus, the number of constraints will scale to $O(T^2)$ if $|R|$ is large which is still difficult to optimize. In the following we consider approximating Eq. (2) by reducing the number of constraints from $O(T^2)$ to $O(T)$.

Note that the relevant-irrelevant label pairs cost the largest number of comparisons. As an optimization objective, many of the comparisons may be redundant. Our basic idea is to use fewer comparisons to approximate them. According to [49], we get the following theorem using our notations.

Theorem 1. *Let $P(l \in R)$ and $P(l \in \overline{R})$ denote the probability that a label l is relevant or irrelevant, respectively. $\mathbb{E}[A]$ is event A's expectation. Then we have*

$$\mathbb{E}\left[\sum_{l_t \in R}\sum_{l_s \in \overline{R}} \frac{\ell_{t,s}}{|\mathcal{B}(t)| \times |\mathcal{B}(s)|}\right] \leq \frac{\mathbb{E}[\sum_{l_t \in R} \ell_{t,\Theta}]}{P(l_t \in R)T} + \frac{\mathbb{E}[\sum_{l_s \in \overline{R}} \ell_{\Theta,s}]}{P(l_s \in \overline{R})T}. \tag{9}$$

Theorem 1 shows that the relevant-irrelevant label pairs can be approximated (in expectation) by the sum of weighted relevant-threshold and irrelevant-threshold pairs, dramatically reducing the number of compared label pairs from $|R|(T-|R|)$ to T. If we use $|R|/T$ and $|\bar{R}|/T$ as estimations of $P(l \in R)$ and $P(l \in \bar{R})$ respectively, an approximation of the righthand side of Eq. (9) can be given. Detailed proof is similar to those in [49] and we omit it here.

Next we will consider simplifying the number of comparisons within relevant labels. Our basic idea is to approximate comparisons between every two relevant labels by a weighted sum of comparisons between every relevant label and its immediate follower. Now the number of compared relevant labels' pairs reduces from $|R|(|R|-1)/2$ to $|R|$.

Theorem 2. *Denote r_i as the index of the ith ranked relevant label, if $\mu_i \geq i(|R|-i)$, we have*

$$\sum_{l_i \in R}\sum_{l_j \in R, j \in \prec_{\mathbf{x}}(i)} \ell_{i,j} \leq \sum_{i=1}^{|R|-1} \mu_i \ell_{r_i, r_{i+1}}. \tag{10}$$

To prove Theorem 2, we first give the following lemma,

Lemma 1. *The accumulated pairwise comparison loss between a relevant label and all labels ranked in front of it has an upper bound as*

$$\sum_{i=1}^{k} \ell_{r_i, r_{k+1}} \leq \sum_{i=1}^{k} i\ell_{r_i, r_{i+1}}. \tag{11}$$

Proof. Assume the left hand side of Eq. (11) equals z, $0 \leq z \leq k$. We want to prove that there exists at least one i, $0 \leq i \leq k-(z+i)$, such that the $(z+i)$th relevant label is ranked incorrectly compared to the $(z+i+1)$th relevant label. We prove this statement by contradiction.

Assuming such kind of relevant label pair does not exist, i.e., $\forall 0 \leq i \leq k-(z+i)$, all $(z+i)$th relevant labels are ranked correctly compared to the $(z+i+1)$th relevant label. Ranking error occurs only within the first $z-1$ relevant labels. Thus $z = \sum_{i=1}^{k} \ell_{r_i, r_{k+1}} = \sum_{i=1}^{z-1} \ell_{r_i, r_{k+1}} \leq z-1$. Because $z \leq z-1$ is impossible, by contradiction, there exists at least one i, $0 \leq i \leq k-(z+i)$, such that the $(z+i)$th relevant label is ranked incorrectly compared to its immediate follower, i.e., the $(z+i+1)$th relevant label.

Without losing generality, assume the z'th relevant label is ranked incorrectly compared to the $(z'+1)$th, $z' \geq z$. Then we have $\sum_{i=1}^{k} i\ell_{r_i, r_{i+1}} \geq z'\ell_{r_{z'}, r_{z'+1}} = z' \geq z$. Thus we finish the proof. □

Proof of Theorem 2. We first rewrite the left hand side of Eq. (10) in Theorem 2 as

$$\sum_{l_i \in R_i}\sum_{l_j \in R, j \prec_{\mathbf{x}}(i)} \ell_{i,j} = \sum_{k=1}^{|R|-1}\sum_{i=1}^{k} \ell_{r_i, r_{k+1}}. \tag{12}$$

Based on Lemma 1, we have

$$\begin{aligned} \sum_{k=1}^{|R|-1}\sum_{i=1}^{k} \ell_{r_i, r_{k+1}} &\leq \sum_{k=1}^{|R|-1}\sum_{i=1}^{k} i\ell_{r_i, r_{i+1}} \\ &= \sum_{i=1}^{k}\sum_{k=1}^{|R|-1} i\ell_{r_i, r_{i+1}} = \sum_{i=1}^{k} i(|R|-1)\ell_{r_i, r_{i+1}}. \end{aligned} \tag{13}$$

We can finish the proof by combining Eqs. (12) and (13).□

According to Theorems 1 and 2, one can approximate the objective function in Eq. (1) with

$$\sum_{l_i \in R} \frac{\ell_{i,\Theta}}{2|\mathcal{B}(i)|} + \sum_{l_j \in \overline{R}} \frac{\ell_{\Theta,j}}{2|\mathcal{B}(j)|} + \sum_{i=1}^{|R|-1} \frac{(i(|R|-i)\ell_{r_i, r_{i+1}})}{2|R|(|R|-1)}, \tag{14}$$

in which the number of constraints scales to $O(T)$. Eq. (14) can be addressed via the same optimization techniques as Eq. (5). We refer to this new algorithm as ProSVM-A (ProSVM Approximation).

4.3 Computational Complexity

In this section, we analyze the computational complexity of ProSVM and ProSVM-A. We first define the notations. Let $\bar{r} \leq T$ be the average number of relevant labels per instance, and $\bar{d} \leq d$ the average number of non-zero features per instance. Assume further the iterations to solve Eq. (4) is K_1 and the number of outer iterations in ProSVM is K_2. In the following, we first consider the computational complexity of solving Eq. (4) without using ADMM and then derive the computational complexity using ADMM.

To solve Eq. (4), we adapt the state-of-the-art SVM solver LIBLINEAR [50], whose computational complexity is linear in the number of dual variables, non-zero entries per instance and number of iterations. Through the definition of $\mathbf{A}$ in Eq. (3), we know that the number of non-zero entries per row of $\mathbf{A}$ is $2\bar{d}$ and there are totally $O(n\bar{r}T)$ rows in $\mathbf{A}$. Thus the time complexity to solve Eq. (4) directly using LIBLINEAR would be $O(n\bar{d}\bar{r}TK_1)$. If we further use ADMM in ProSVM which divides the whole data into Z folders to release the storage burden, the time complexity would be $O(n\bar{d}\bar{r}TK_1K_2)$ as updating $\bar{\mathbf{w}}_0$ and α cost only linear time. If

TABLE 2
The Computational Complexity of ProSVM and ProSVM-A w/o Using ADMM and Parallelization with up to Z Cores, where n is the Number of Instances, T is the Number of Labels, $\bar{d}$ is the Average Number of Non-Zero Features per Instance, and $\bar{r}$ is the Average Number of Relevant Labels per Instance

	No ADMM	Using ADMM	
		No parallel	With parallel
ProSVM	$O(n\bar{d}\bar{r}TK_1)$	$O(n\bar{d}\bar{r}TK_1K_2)$	$O((n\bar{d}\bar{r}TK_1K_2/Z)$
ProSVM-A	$O(n\bar{d}TK_1)$	$O(n\bar{d}TK_1K_2)$	$O(n\bar{d}TK_1K_2/Z)$

K_1 is the number of iterations of solving Eq. (6), and K_2 is the total number of outer iterations.

we can use up to Z cores to parallelize, then the computational complexity would be $O(n\bar{d}\bar{r}TK_1K_2/Z)$.

For ProSVM-A, rows in $\mathbf{A}$ would be reduced to $O(nT)$. Thus the computational complexity without using ADMM is $O(n\bar{d}TK_1)$. When using ADMM, the computational complexity is $O(n\bar{d}TK_1K_2)$. With the power of parallelization, the computational complexity would be further reduced to $O(n\bar{d}TK_1K_2/Z)$. The results on computational complexity are summarized in Table 2.

As analyzed in [50], K_1 would be $O(\log(1/\epsilon_1))$ if we need to get an ϵ_1-optimal solution to Eq. (6). According to [51], K_2 would be $O(1/\epsilon_2)$ if we need the ϵ_2-optimal solution. Although theoretically the $O(1/K_2)$ does not converge fast, in practice, a good approximate solution is sufficient [45]. In our experiment, the maximal iteration is simply set to 100 and empirical results showing how ProSVM will converge within 100 iterations are showed in Fig. 2, validating the effectiveness of our proposal. For the details of the datasets, please refer to Section 6.

From Table 2, we can see that the computational complexity of ProSVM and ProSVM-A are linear in the number of parameters, and the time complexity can be further improved if the data is sparse ($\bar{d} \ll d$), or using parallelization. Comparing ProSVM and ProSVM-A, we found that when the number of relevant labels is relatively small (i.e., $\bar{r} \ll T$), using the two algorithms will result in no difference asymptotically in computational complexity, although in practice, it still involves difference. However, when $\bar{r}$ is relatively large, using ProSVM-A will save a lot of computational cost compared to ProSVM.

5 PROSVM WITH PARTIAL LABELS

In this section, we extend ProSVM to handle the partial labels in multi-label training data. Here we consider the case when the annotation information is uniformly random missing as in [14], [20], [35] and defer the non-uniformly missing case to future work.

Since we are dealing with the problem when relevant labels are ranked, we assume for those observed relevant labels, we also have the partial ranking information of them. Under this scenario, the large margin surrogate convex loss will become

$$\min_{\mathbf{g}} \lambda \sum\nolimits_{i=1}^{n} \tilde{L}_P(\mathbf{x}_i, R_i, \prec, \mathbf{g}, \Omega_i) + \mathrm{Regularizer}(\mathbf{g}), \tag{15}$$

where

$$\tilde{L}_P(\mathbf{x}_i, R_i, \prec, \mathbf{g}, \Omega_i) = \sum_{l_t \in (R_i \cap \Omega_i) \cup \{\Theta\}} \sum_{s \in (\prec_{\mathbf{x}_i}(t) \cap \Omega_i)} \frac{1}{4h_{s,t}} (1 + g_s(\mathbf{x}_i) - g_t(\mathbf{x}_i))_+,$$

and $\Omega_i \subset [m]$ is $\mathbf{x}_i$'s indices set of observed labels.

Assume in the same way as Section 4 $g_t(\mathbf{x}) = \mathbf{w}_t^\top \mathbf{x}$, $t \in \{1, \ldots, T\} \cup \{\Theta\}$ and $\mathrm{Regularizer}(\mathbf{g}) = \frac{1}{2}\sum_{t \in \{1,\ldots,T\}\cup\{\Theta\}} \|\mathbf{w}_t\|^2$. Let $\mathbf{w} \triangleq [\mathbf{w}_1; \ldots; \mathbf{w}_T; \mathbf{w}_\Theta]$ and D be the training set, Eq. (15) can be cast into the following optimization problem

$$\min_{\mathbf{w}} F_P(\mathbf{w}, D) \triangleq \frac{1}{2}\|\mathbf{w}\|^2 + \lambda \mathbf{C}_P^\top \Big(\mathbf{1}_{\hat{p}} - \mathbf{A}_P \mathbf{w}\Big)_+, \tag{16}$$

where $\hat{p}$ is the total number of constraints introduced by observed labels in all $\Omega_i, \forall i$. $\mathbf{A}_P$ is defined in the same way as $\mathbf{A}$ but considering necessary comparisons between only observed labels.

One crucial difference between Eqs. (16) and (4) is that $\mathbf{C}_P$ in Eq. (16) is *unknown* since we do not have any idea how many relevant and irrelevant labels are presented in one

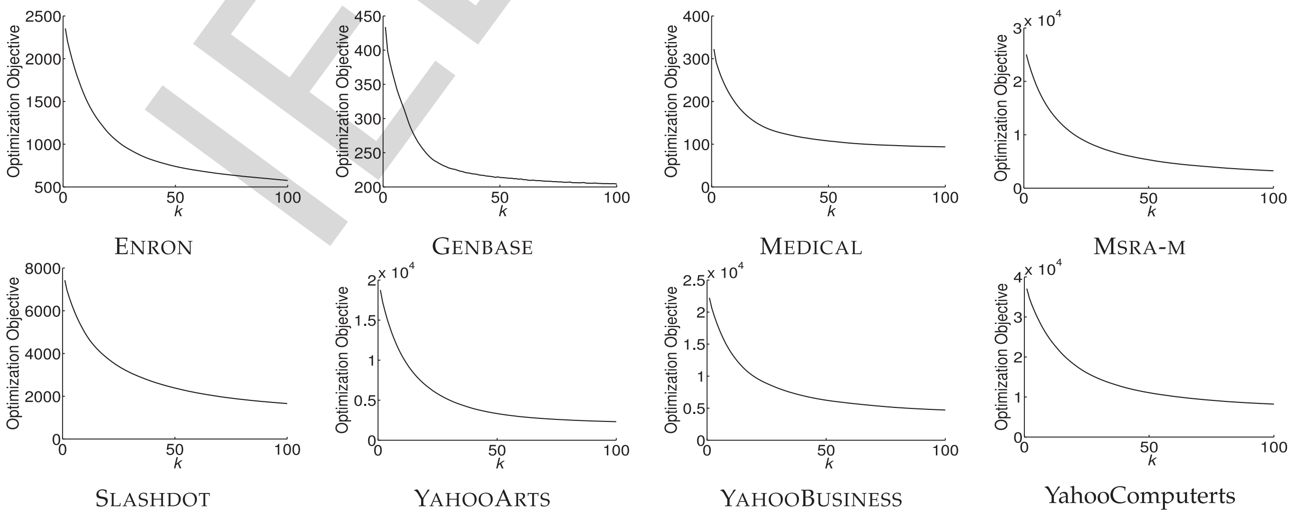

Fig. 2. Convergence results of ProSVM on 8 representative datasets in 100 iterations. The Y-axis is the optimization objective $\mathbb{L}$, and the X-axis is the kth iteration.

instance given partial labels, thus we cannot use the same optimization procedure as Eq. (4) directly. Before using the same optimization strategy, we need to first estimate $\mathbf{C}_P$. To estimate $\mathbf{C}_P$, there are a bunch of ways. The most simple and straightforward one is to ignore the weights of different label pairs and simply set $\mathbf{C}_P$ to be an all-one matrix. We call this method ProSVM-PI (ProSVM for Partial Labels with Identity Weights) as it sets all weights to be identical.

Considering that the labels are observed uniformly at random, another way to estimate $\mathbf{C}_P$ is to get an unbiased estimation of it. More specially, by assuming the annotation information is uniformly random missing with probability $1-\omega\%$, we can have an unbiased estimation of the real number of relevant and irrelevant labels: when the observed labels' indices are in $|\Omega_i|$, the estimated number of relevant labels and irrelevant labels are $|\Omega_i \cap R_i|/\omega_i$ and $|\Omega_i \cap \bar{R}_i|/\omega_i$ respectively. We call this method ProSVM-PU (ProSVM for Partial Labels with Unbiased Estimation).

We can also estimate $\mathbf{C}_P$ using domain knowledge or asking domain experts to provide the real number of relevant labels directly. Note that this is the best result that our proposed ProSVM-P methods can achieve, and we call this method ProSVM-PD (ProSVM for Partial Labels with Domain Knowledge/Experts).

Note that ProSVM-A approximately optimizes PRO LOSS using the comparisons between relevant labels and their immediate followers. When the labels are partial, it is nontrivial for the algorithm to determine whether the label ranked behind is its immediate follower or not, thus making the ProSVM-A method unsuitable for the partial label case. Furthermore, when the labels are partial, we have a much smaller number of pairwise comparisons, making the ProSVM-A method unnecessary in most cases. Parallelization of ProSVM-P algorithms could probably be solved by recent work on distributed learning with incomplete data [52] and we plan to study such kind of possibility in future work.

6 EXPERIMENTS WITH FULL LABELS

Our proposals are compared to a number of algorithms. First, we compare with classical multi-label methods which cannot handle relevant labels' ranking. For small datasets, these methods include PC [30], RankSVM [11], BSVM [8], ML-kNN [40] and BoosTexter [39]. We use two implementations of PC, i.e., PCn and PC0. In PCn, Perceptron stops after n rounds while in PC0, it stops when no error occurs. Next, we extend these methods to take the relevant labels' ranking into consideration, i.e., after we do classification, we further use the pairwise label ranking method [53] to rank the relevant labels. In this way, we get two variants of PC, namely PCnR and PC0R, and one extension of RankSVM, named as RankSVM-R. Third, we compare with GMLC [31] which considers multiple degrees of label relevancies. To run GMLC, the number of relevance levels is fixed to be $\max_{i=1}^n(|R_i|+1)$, and the ith relevant label is assigned to the ith level while the irrelevant labels are assigned to the $\left(\max_{i=1}^n(|R_i|+1)\right)$th level. Finally we will compare with two extreme multi-label learning algorithms, PD-Sparse [44] and PfastreXML [6]. For large datasets, we will compare with four recent proposed methods including GLOCAL [24], LIMO [38], MLGT [54] and genEML [13]. Since these methods are designed recently targeted at more large datasets, thus we will conduct these compared methods on large data only.

The setups of our proposals and compared methods are as follows. For ProSVM and RankSVM, the regularization parameter is selected from $\{2^{-10}, 2^{-8}, \ldots, 2^8, 2^{10}\}$ by ten-fold cross validation on small data sets, and simply set as 1 on two large data sets. For BSVM, the SVM is implemented by LIBSVM [55] package with parameters selected in the same way as RankSVM. For ML-kNN, we use the parameter setting recommended by [40]. For BoosTexter, we use the version AdaBoost.MH [39]. For ProSVMs η is fixed to 0.1. The split number Z is fixed to $(p \times d)/10^7$ where p is the number of constraints in Eq. (3). Hence, the memory requirement of ProSVM is low and applicable for most personal computers. For PD-Sparse and PfastreXML, we have conducted extensive parameter selection recommended in the original paper and report the best results on test data. For GLOCAL we use the default parameter. For LIMO, we use the version of optimizing Hamming Loss. For MLGT and genEML, we use the recommended parameters.

6.1 Data with Synthetic Ranking of Relevant Labels

Most of the multi-label datasets do not contain ranking information for relevant labels, thus in this part, we consider synthesizing the relevant labels' ranking for real multi-label data, and evaluate our proposal on these datasets. To synthesize a reasonable ranking for relevant labels, we first consider employing several human annotators in a crowdsourcing way to give the relevant labels' rankings, and then aggregating by averaging the results and giving the final ranking for all the relevant labels. However, employing people to label ten or more multi-label datasets will take a lot of money, thus in this part, we consider simulating this process by employing several agents replacing the human annotators, and employ real people to construct a real data set in Section 6.2. Inspired by that existing multi-label learning methods can give each label a real value indicating the algorithm's confidence in predicting the label as relevant, we will use existing algorithm as "pseudo human annotator". We then evaluate our proposal and compared methods on the synthetic datasets to see whether our proposal can fit the oracle of pseudo annotators. More specially, for smaller data sets we synthesize the relevant labels' ranking by automatically running 3 state-of-the-art multi-label methods [56], [57], [58]. Each predicts a real-valued score for each label, and then we obtain the ranking of relevant labels by sorting the aggregated scores. By this approach, a broad range of 19 datasets which cover diverse domains, e.g., *music*, *biology*, *image* and *text*, are studied. The numbers of samples vary from 590 to 5,000, the numbers of dimensionality vary from 72 to 1,449 and the numbers of labels vary from 5 to 53. These datasets have been widely used and public available.[2] For large data sets, we use *Bibtex* and *Delicious*[3] containing 7395 and 16105 instances, 159 and 983 labels respectively. We run FastXML [43] and use the

2. The EMOTIONS, ENRON, GENBASE, MEDICAL, SCENE, and YEAST datasets are publicly available at http://mulan.sourceforge.net/datasets.html, the IMAGE and 11 YAHOO datasets are available at http://cse.seu.edu.cn/people/zhangml/Resources.htm, and the SLASHDOT data are available at http://meka.sourceforge.net.

3. These two data sets are available at http://manikvarma.org/downloads/XC/XMLRepository.html

TABLE 3
Comparison Results on PRO Loss for Data with Synthetic Ranking of Relevant Labels on Small Data

Data set	ProSVM	ProSVM-A	PCn	PCnR	PC0	PC0R	RSVM	RSVM-R	BSVM	MLk	Btx	GMLC	PD-S	PfXML
emotions	.1982	.1980	.3557	.3509	.2821	.2641	.2159	.2110	**.1814**	.2210	.2397	.2255	.2493	.4928
enron	**.1343**	.1349	.3015	.3032	.3143	.3031	.1507	.1587	.2136	.2533	.2121	.3733	.2887	.4868
genbase	**.0022**	.0023	.2544	.2544	.0511	.0489	.0057	.0074	.0232	.0181	.0049	.0113	.0233	.3696
image	.1604	.1595	.2755	.2738	.2481	.2518	.1992	.2009	.1601	.1914	.1737	.2150	.4036	**.0069**
medical	**.0569**	.0600	.2769	.2769	.2038	.1998	.0890	.0895	.1265	.1647	.0838	.1684	.1355	.4153
scene	**.0994**	.1010	.2829	.2840	.2710	.2713	.1198	.1243	.1132	.1228	.1081	.1405	.1692	.2743
slashdot	**.1153**	.1180	.2877	.2877	.2781	.2766	.1674	.1676	.1892	.2944	.1793	.3632	.3961	.4387
YahooArts	**.1503**	.1509	.3176	.3179	.3062	.3060	.2287	.2304	.2519	.3067	.2474	.3888	.4295	.4674
YahooBusiness	.0601	.0600	.2673	.2673	.1713	.1713	.0832	.0845	.1123	.0921	.0912	.1206	.4274	**.0192**
YahooComputers	**.0971**	.0993	.2861	.2864	.1599	.1599	.1669	.1675	.2044	.2073	.1852	.2695	.4141	.5120
YahooEducation	.1114	**.1102**	.2951	.2939	.1830	.1828	.2057	.2064	.2182	.2479	.2264	.3228	.4172	.5169
YahooEntertainment	.1192	**.1188**	.2955	.2933	.1677	.1674	.1866	.1875	.1870	.2419	.2064	.3118	.4103	.6088
YahooHealth	**.0898**	.0930	.3045	.2961	.1553	.1547	.1467	.1494	.2280	.2044	.1619	.2933	.4340	.5094
YahooRecreation	.1544	**.1524**	.3026	.3018	.2800	.2803	.2244	.2252	.2162	.3045	.2438	.3715	.4140	.4550
YahooReference	.0934	**.0920**	.2779	.2779	.1480	.1485	.1565	.1566	.1914	.2296	.1783	.3092	.3932	.5451
YahooScience	.1389	**.1386**	.2985	.2988	.2154	.2157	.2176	.2190	.2400	.2628	.2480	.3297	.4132	.4649
YahooSocial	**.0858**	.0890	.2853	.2856	.1630	.1626	.1356	.1369	.1663	.1648	.1542	.2299	.4035	.5165
YahooSociety	.1515	**.1503**	.3114	.3111	.2654	.2632	.2016	.2020	.2279	.2280	.2308	.2993	.4250	.5646
yeast	**.1853**	.1867	.3472	.3406	.4177	.4141	.1931	.2557	.2094	.2338	.2548	.2326	NaN	.5082
R-Total	32	32	215	211	150	142	79	102	114	151	113	190	227	237

Each entry presents the PRO Loss; the best result of each dataset is bold. For Image and Slashdot that have not provided training/testing splits, 10-CV is conducted and average performances are recorded. For other datasets, we use the provided training/testing splits. The last row R-total presents the sum of ranks; the smaller the R-total, the better the overall performance. (RSVM(-R): RankSVM(-R); MLk: MLkNN; Btx: BoosTexter; PD-S: PD-Sparse; and PfXM: PfastreXML)

predicted real-valued scores for each label to rank the relevant labels.

The results on small data sets are shown in Table 3. As can be seen, ProSVMs perform superior compared to state-of-the-art methods. In particular, ProSVM achieves the best results on 13 over 19 datasets followed by ProSVM-A achieving the best results on the remaining 6. The results on large data sets are shown in Table 4. genEML fails to give any result on Bibtex data so we use an NaN. We can see that, compared with more recent proposed methods, our proposals are still superior on large datasets. Specially, the two proposed methods perform the best in all compared methods. LIMO targeting at Hamming Loss performs the second on Bibtex but not that good on Delicious.

6.2 Data with Real Ranking of Relevant Labels

In this part, we exploit the strategy of employing several human annotators, and provide the first real-world data Msra-m with relevant labels' ranking. Specifically, we use a subset of the widely-used Msra dataset [59], which contains 1868 images, with 899 features for each image. There are 19 candidate labels, while each image contains 1 to 11 relevant labels. We use a crowdsourcing platform to spread the task to human annotators, asking them to provide the ranking of all the relevant labels. Then we average all the obtained results and provide the real-world dataset. In our experiment, 10-CV is conducted to give the average results.

TABLE 4
Comparison Results on PRO Loss for Data with Synthetic Ranking of Relevant Labels on Large Data

Data Set	P-SVM	P-SVM-A	GLOCAL	LIMO	MLGT	genEML
Bibtex	**0.1499**	0.1529	0.3456	0.1949	0.3140	NaN
Delicious	0.2139	**0.2030**	0.3701	0.3365	0.4415	0.3641

Each entry presents the PRO Loss; the best result of each dataset is bold. 10-CV is conducted and average performances are recorded. (P-SVM(-A): ProSVM(-A))

The experimental results are shown in Table 5. Since PD-Sparse fails to give any result, we just use an NaN to denote its evaluation. As can be seen, ProSVMs perform significantly better than all other compared methods. Demonstration of two concrete examples in the Msra-m dataset is shown in Table 6. From the last two columns, we can see that compared to state-of-the-art algorithms, our proposal can not only give a good classification of all the labels, but also a good ranking of relevant labels.

6.3 Performance on Other Measurements

We have shown our algorithm can perform well measured by PRO Loss. Although our proposed algorithm are targeted at PRO Loss, we may also expect that it will not perform bad on other measurement. In this part, we will show our proposal can have comparable performance with existing methods on classical multi-label measurements without considering relevant labels' ranking. We will use the same 19 small datasets in Section 6.1. Here to give a fair comparison,

TABLE 5
Results (mean±std) on Msra-m with Real Ranking of Relevant Labels

Method	PRO Loss	Method	PRO Loss
ProSVM	**.2536** ± **.0107**	RSVM	.2955 ± .0145
ProSVM-A	**.2587** ± **.0115**	RSVM-R	.2656 ± .0117
PCn	.3754 ± .0406	BSVM	.2913 ± .0070
PCnR	.3469 ± .0420	MLkNN	.3228 ± .0099
PC0	.3149 ± .0107	Btx	.2957 ± .0112
PC0R	.3040 ± .0090	GMLC	.3052 ± .0130
PD-Spar	NaN	P-XML	.2802 ± .0037

The best performance and its comparable ones (pairwise t-test at 95 percent confidence) are bold. (RSVM(-R): RankSVM(-R); Btx: BoosTexter; PD-Spar: PD-Sparse; and P-XML: PfastreXML.)

TABLE 6
Demonstration of the Prediction on Image Annotations Tasks

Image	A.	Prediction	C	R
	P	*people > woman > clothing > leaf > hat*	1	5
	A	*woman > people > leaf > clothing > hat*	1	7
	n	*NaN*	6	10
	nR	*NaN*	6	9
	0	*leaf > clothing > people > building > jungle > woman*	4	10
	0R	*people > leaf > clothing > woman > building > jungle > flower*	5	8
	R	*leaf > people > clothing > hat > woman > jungle > door > car > flower > building > animal > flag > city*	7	10
	RR	*woman > people > clothing > hat > animal > car > building > flower > flag > city > door > jungle > leaf*	7	6
	B	*leaf > woman > people > clothing > sky*	3	9
flag > people > woman > clothing> leaf > hat	M	*leaf > people > jungle > sky > clothing*	5	10
	T	*people > leaf > woman > clothing > water > sky > cloud*	5	7
	G	*woman > people > clothing > hat > leaf > jungle*	2	6.5
	X	*sky > animal*	8	7.5
	P	*people > clothing > hat > leaf > building > jungle > woman*	4	6
	A	*people > clothing > leaf > hat > nature > jungle > building*	4	9
	n	*NaN*	7	7
	nR	*people*	6	7
	0	*people > jungle > hat > clothing > leaf*	4	8
	0R	*people > jungle > clothing > nature > hat*	4	10
	R	*people > clothing > jungle > hat > leaf > building > flag > city > car > woman > flower*	8	6
	RR	*people > woman > clothing > flower > car > city > building > flag > hat > leaf > jungle*	8	4
	B	*people > jungle > leaf > clothing > hat*	4	10
people > woman > clothing> water > jungle > leaf > nature	M	*leaf > sky > nature > cloud > people*	6	16
	T	*leaf > people > jungle*	4	12
	G	*people > jungle > leaf > sky > nature*	4	11
	X	*sky > animal*	9	10.5

Below the image is the ground truth. "A." denotes the abbreviation of algorithms (P: ProSVM; A: ProSVM-A; n: PCn; nR: PCnR; 0: PC0; 0R: PC0R; R: RankSVM; RR: RankSVM-R; B: BSVM; M: MLkNN; T: BoosTexter; G: GMLC; and X: PfastreXML). The right two columns are the "number of wrongly classified labels" denoted by C and "number of wrongly ranked relevant label pairs" denoted by R. The smaller the value, the better the performance.

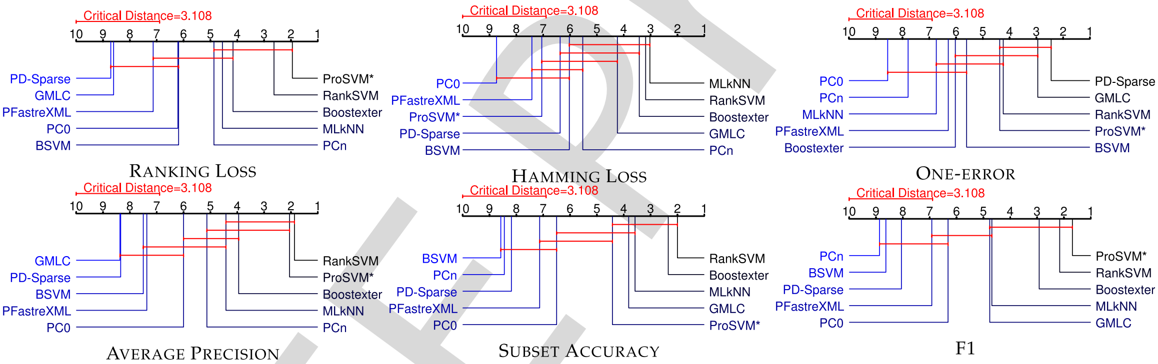

Fig. 3. Comparison of ProSVM* with nine other multi-label methods on classical measurements to show that ProSVM* can get comparable performance. The average rank of classifiers across multiple datasets are shown on the number line. Groups of classifiers that are not significantly different are connected by red line.

TABLE 7
Comparison Showing the Training Time in Seconds on the Algorithm Denoted by Column and Dataset Denoted by Row

Dataset	ProSVM	ProSVM-A	PCn	PC0	RankSVM	BSVM	MLknn	Boostexter	GMLC	PD-Sparse	PfastreXML
emotions	8×10^{-1}	1×10^{0}	2×10^{1}	2×10^{2}	2×10^{0}	4×10^{0}	4×10^{-1}	6×10^{0}	3×10^{0}	6×10^{-2}	3×10^{-1}
enron	1×10^{3}	7×10^{2}	3×10^{3}	2×10^{4}	2×10^{2}	1×10^{1}	1×10^{1}	1×10^{2}	1×10^{3}	1×10^{0}	1×10^{0}
genbase	5×10^{1}	3×10^{1}	5×10^{2}	1×10^{2}	5×10^{0}	3×10^{-1}	2×10^{0}	7×10^{1}	5×10^{1}	8×10^{-3}	2×10^{-1}
image	1×10^{1}	6×10^{0}	6×10^{1}	1×10^{3}	7×10^{0}	1×10^{2}	1×10^{1}	9×10^{1}	4×10^{1}	6×10^{-1}	5×10^{0}
medical	1×10^{2}	1×10^{2}	1×10^{3}	1×10^{1}	9×10^{0}	1×10^{-1}	5×10^{0}	6×10^{1}	8×10^{1}	2×10^{-2}	2×10^{-1}
scene	4×10^{0}	2×10^{0}	4×10^{1}	6×10^{2}	5×10^{0}	2×10^{1}	5×10^{0}	7×10^{1}	2×10^{1}	9×10^{-1}	3×10^{0}
slashdot	3×10^{2}	2×10^{2}	2×10^{3}	6×10^{3}	6×10^{1}	1×10^{1}	1×10^{2}	5×10^{2}	6×10^{2}	1×10^{-1}	2×10^{0}
Y.Arts	2×10^{2}	1×10^{2}	8×10^{2}	5×10^{3}	7×10^{1}	5×10^{1}	2×10^{1}	1×10^{2}	5×10^{2}	2×10^{-1}	1×10^{0}
Y.Heal.	4×10^{2}	2×10^{2}	1×10^{3}	5×10^{3}	7×10^{1}	4×10^{1}	3×10^{1}	2×10^{2}	5×10^{2}	2×10^{-1}	1×10^{0}
Y.Sci.	8×10^{2}	3×10^{2}	2×10^{3}	2×10^{3}	8×10^{1}	2×10^{1}	3×10^{1}	2×10^{2}	6×10^{2}	3×10^{-1}	2×10^{0}
Y.Bus.	3×10^{2}	1×10^{2}	1×10^{3}	9×10^{3}	7×10^{1}	3×10^{1}	2×10^{1}	1×10^{2}	6×10^{2}	2×10^{-1}	1×10^{0}
Y.Com.	5×10^{2}	2×10^{2}	2×10^{3}	2×10^{4}	7×10^{1}	3×10^{1}	3×10^{1}	2×10^{2}	1×10^{3}	2×10^{-1}	1×10^{0}
Y.Edu.	3×10^{2}	2×10^{2}	1×10^{3}	2×10^{3}	7×10^{1}	5×10^{1}	2×10^{1}	2×10^{2}	5×10^{2}	2×10^{-1}	1×10^{0}
Y.Ent.	2×10^{2}	1×10^{2}	7×10^{2}	4×10^{3}	5×10^{1}	2×10^{1}	3×10^{1}	2×10^{2}	4×10^{2}	2×10^{-1}	1×10^{0}
Y.Rec.	2×10^{2}	2×10^{2}	7×10^{2}	6×10^{3}	4×10^{1}	2×10^{1}	3×10^{1}	2×10^{2}	6×10^{2}	2×10^{-1}	1×10^{0}
Y.Ref.	3×10^{2}	2×10^{2}	2×10^{3}	9×10^{3}	5×10^{1}	1×10^{1}	4×10^{1}	2×10^{2}	4×10^{2}	2×10^{-1}	2×10^{0}
Y.Social	6×10^{2}	5×10^{2}	3×10^{3}	6×10^{2}	7×10^{1}	2×10^{1}	4×10^{1}	3×10^{2}	1×10^{3}	3×10^{-1}	2×10^{0}
Y.Society	3×10^{2}	2×10^{2}	1×10^{3}	4×10^{3}	6×10^{1}	4×10^{1}	3×10^{1}	2×10^{2}	8×10^{2}	2×10^{-1}	2×10^{0}
yeast	1×10^{1}	3×10^{0}	2×10^{2}	0×10^{0}	3×10^{1}	6×10^{1}	5×10^{0}	3×10^{1}	1×10^{2}	NaN	9×10^{-1}

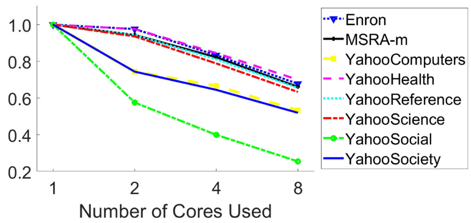

Fig. 4. Comparison on the time of ProSVM on multiple cores. The X-axis is the number of cores, and the Y-axis is the ratio dividing the running time on multiple cores by the running time on 1 core.

our proposal is evaluated by neglecting the relevant labels' ranking. Specifically, a simpler loss function without comparing pairs of relevant labels is used for ProSVMs, and the same optimization techniques are applied. We call our new variants as ProSVM*. Note that PCnR, PC0R and RankSVM-R could not be compared since they require the ranking information. For GMLC, two relevance levels, i.e., relevant and irrelevant, are used.

We use the Critical Distance (CD) Diagram [60] to show the ProSVM*'s overall performance on 19 datasets compared to 9 methods. The CD Diagram widely used in previous multi-label studies [61], [62], shows the average rank, as well as the Nemnyi test results. The CD Diagram on 6 measurements are shown in Fig. 3. As can be seen, our proposal still performs highly competitive on existing criteria. Specifically, ProSVM*'s performance is comparable to the best one by state-of-the-art methods on Ranking Loss, One-error, Average Precision, Subset Accuracy and F1 criteria, and achieves comparable performance to most algorithms on Hamming

TABLE 8
Results (mean±std) on Small Multi-Label Datasets with Partial Labels where MC can Finish within 24 Hours, Measured by Pro Loss

Data	Algo.	$\omega = 10\%$	$\omega = 20\%$	$\omega = 30\%$	$\omega = 40\%$
emotions	BSVM	.2560 ± .0009	.2350 ± .0009	.2163 ± .0007	.2088 ± .0008
	MC-b	.4100 ± .0006	.4230 ± .0004	.4276 ± .0006	.4257 ± .0005
	MC-1	.4140 ± .0008	.4159 ± .0003	.4200 ± .0005	.4221 ± .0004
	Maxide	.3169 ± .0006	.2949 ± .0005	.2858 ± .0003	.2865 ± .0004
	ProSVM-PI	.2657 ± .0010	.2378 ± .0006	.2230 ± .0007	.2145 ± .0008
	ProSVM-PU	.2656 ± .0009	.2298 ± .0009	.2093 ± .0004	.1990 ± .0007
	ProSVM-PD	**.2433 ± .0005**	**.2106 ± .0006**	**.1878 ± .0004**	**.1861 ± .0004**
genbase	BSVM	.1051 ± .0007	.0724 ± .0006	.0533 ± .0003	.0484 ± .0003
	MC-b	.2769 ± .0010	.2613 ± .0007	.2631 ± .0000	.2556 ± .0003
	MC-1	.2962 ± .0011	.2759 ± .0005	.2799 ± .0004	.2996 ± .0002
	Maxide	.1268 ± .0003	.1075 ± .0001	.1020 ± .0002	.1058 ± .0002
	ProSVM-PI	.0312 ± .0005	.0147 ± .0001	.0083 ± .0000	.0070 ± .0000
	ProSVM-PU	.0310 ± .0004	.0136 ± .0000	.0082 ± .0000	.0071 ± .0000
	ProSVM-PD	**.0302 ± .0005**	**.0137 ± .0000**	**.0081 ± .0000**	**.0076 ± .0000**
image	BSVM	.2327 ± .0001	.2175 ± .0001	.2103 ± .0001	.2049 ± .0002
	MC-b	.3995 ± .0000	.4016 ± .0001	.4009 ± .0001	.4028 ± .0001
	MC-1	.3694 ± .0001	.3642 ± .0003	.3643 ± .0002	.3614 ± .0001
	Maxide	.2899 ± .0001	.2877 ± .0001	.2843 ± .0000	.2862 ± .0000
	ProSVM-PI	.2412 ± .0001	.2298 ± .0001	.2226 ± .0001	.2162 ± .0001
	ProSVM-PU	.2377 ± .0001	.2222 ± .0001	.2060 ± .0001	.1921 ± .0001
	ProSVM-PD	**.2154 ± .0001**	**.1954 ± .0001**	**.1867 ± .0001**	**.1821 ± .0001**
medical	BSVM	.2727 ± .0012	.2007 ± .0012	.1685 ± .0003	.1485 ± .0001
	MC-b	.2977 ± .0001	.2861 ± .0000	.2844 ± .0000	.2815 ± .0000
	MC-1	.2916 ± .0001	.2830 ± .0000	.2787 ± .0000	.2757 ± .0000
	Maxide	.1652 ± .0002	.1576 ± .0000	.1549 ± .0000	.1536 ± .0000
	ProSVM-PI	.1566 ± .0007	.1185 ± .0004	.0966 ± .0002	.0895 ± .0001
	ProSVM-PU	.1535 ± .0005	.1094 ± .0002	.0786 ± .0001	.0673 ± .0001
	ProSVM-PD	**.1286 ± .0004**	**.0926 ± .0002**	**.0712 ± .0001**	**.0613 ± .0001**
scene	BSVM	.1466 ± .0001	.1380 ± .0001	.1349 ± .0001	.1308 ± .0001
	MC-b	.3860 ± .0000	.3824 ± .0001	.3815 ± .0000	.3819 ± .0000
	MC-1	.3179 ± .0006	.3054 ± .0009	.3158 ± .0014	.3027 ± .0007
	Maxide	.2148 ± .0001	.2118 ± .0001	.2102 ± .0001	.2100 ± .0001
	ProSVM-PI	.1561 ± .0001	.1476 ± .0001	.1437 ± .0002	.1379 ± .0001
	ProSVM-PU	.1500 ± .0001	.1388 ± .0001	.1246 ± .0002	.1139 ± .0002
	ProSVM-PD	**.1226 ± .0001**	**.1099 ± .0001**	**.1051 ± .0001**	**.1012 ± .0001**
slashdot	BSVM	.2890 ± .0002	.2499 ± .0001	.2403 ± .0000	.2280 ± .0001
	MC-b	.3263 ± .0000	.3274 ± .0000	.3280 ± .0000	.3271 ± .0000
	MC-1	.3296 ± .0000	.3299 ± .0000	.3286 ± .0000	.3277 ± .0000
	Maxide	.2409 ± .0001	.2332 ± .0001	.2280 ± .0000	.2205 ± .0001
	ProSVM-PI	.2282 ± .0001	.2057 ± .0000	.1962 ± .0000	.1870 ± .0000
	ProSVM-PU	.2270 ± .0001	.1974 ± .0000	.1801 ± .0000	.1641 ± .0000
	ProSVM-PD	**.1908 ± .0001**	**.1667 ± .0000**	**.1507 ± .0001**	**.1389 ± .0001**
yeast	BSVM	.2481 ± .0002	.2337 ± .0001	.2295 ± .0000	.2297 ± .0000
	MC-b	.2870 ± .0002	.2859 ± .0001	.2844 ± .0001	.2805 ± .0001
	MC-1	.2815 ± .0001	.2781 ± .0001	.2726 ± .0000	.2699 ± .0000
	Maxide	.4226 ± .0001	.4062 ± .0001	.3995 ± .0001	.3981 ± .0001
	ProSVM-PI	.2416 ± .0001	.2187 ± .0001	.2116 ± .0001	.2062 ± .0000
	ProSVM-PU	.2360 ± .0001	.2189 ± .0001	.2048 ± .0001	.1961 ± .0000
	ProSVM-PD	**.2195 ± .0001**	**.2007 ± .0000**	**.1919 ± .0001**	**.1873 ± .0001**

"Algo." specifies the name of the algorithms. ω% represents the percentage of observed label assignments in training instances. The best result and its comparable ones (pairwise single-tailed t-tests at 95 percent confidence level) are bold.

TABLE 9
Results (mean±std) on Small Multi-Label Datasets with Partial Labels where MC cannot Finish within 24 Hours, Measured by Pro Loss

Data	Algo.	$\omega = 10\%$	$\omega = 20\%$	$\omega = 30\%$	$\omega = 40\%$
	BSVM	.3184 ± .0006	.2761 ± .0003	.2612 ± .0003	.2501 ± .0002
	Maxide	.2787 ± .0001	.2735 ± .0001	.2760 ± .0000	.2678 ± .0001
enron	ProSVM-MI	.2375 ± .0002	.2130 ± .0001	.2015 ± .0001	.1910 ± .0001
	ProSVM-MU	.2230 ± .0002	.1861 ± .0001	.1701 ± .0001	.1603 ± .0001
	ProSVM-MD	**.1810 ± .0002**	**.1639 ± .0001**	**.1562 ± .0001**	**.1499 ± .0001**
	BSVM	.3225 ± .0000	.3096 ± .0001	.3042 ± .0001	.3022 ± .0000
	Maxide	.3938 ± .0000	.3917 ± .0000	.3954 ± .0000	.3932 ± .0000
Msra-m	ProSVM-MI	.3212 ± .0001	.3053 ± .0001	.2969 ± .0000	.2898 ± .0000
	ProSVM-MU	.3208 ± .0000	.2994 ± .0000	.2885 ± .0000	.2804 ± .0000
	ProSVM-MD	**.3123 ± .0000**	**.2918 ± .0001**	**.2811 ± .0000**	**.2754 ± .0000**
	BSVM	.3074 ± .0003	.2864 ± .0002	.2756 ± .0002	.2711 ± .0002
	Maxide	.2948 ± .0000	.2824 ± .0000	.2728 ± .0000	.2684 ± .0000
YahooArts	ProSVM-MI	.2677 ± .0001	.2564 ± .0001	.2494 ± .0001	.2424 ± .0001
	ProSVM-MU	.2484 ± .0000	.2093 ± .0000	.1780 ± .0000	.1642 ± .0000
	ProSVM-MD	**.1786 ± .0001**	**.1659 ± .0000**	**.1592 ± .0000**	**.1547 ± .0000**
	BSVM	.1554 ± .0001	.1356 ± .0001	.1318 ± .0001	.1269 ± .0002
	Maxide	.2097 ± .0000	.1973 ± .0000	.1885 ± .0000	.1820 ± .0000
YahooBusiness	ProSVM-MI	.0991 ± .0000	.0944 ± .0000	.0924 ± .0000	.0890 ± .0000
	ProSVM-MU	.0823 ± .0000	.0738 ± .0000	.0660 ± .0000	.0618 ± .0000
	ProSVM-MD	**.0684 ± .0000**	**.0642 ± .0000**	**.0610 ± .0000**	**.0584 ± .0000**
	BSVM	.2566 ± .0002	.2249 ± .0001	.2130 ± .0001	.2051 ± .0001
	Maxide	.2485 ± .0000	.2425 ± .0000	.2343 ± .0000	.2258 ± .0000
YahooComputers	ProSVM-MI	.1977 ± .0001	.1837 ± .0000	.1743 ± .0000	.1699 ± .0000
	ProSVM-MU	.1727 ± .0000	.1329 ± .0000	.1208 ± .0000	.1135 ± .0000
	ProSVM-MD	**.1207 ± .0000**	**.1134 ± .0000**	**.1085 ± .0000**	**.1043 ± .0000**
	BSVM	.3606 ± .0004	.3082 ± .0002	.2820 ± .0003	.2684 ± .0003
	Maxide	.2540 ± .0001	.2589 ± .0001	.2501 ± .0000	.2406 ± .0000
YahooEducation	ProSVM-MI	.2376 ± .0000	.2278 ± .0001	.2217 ± .0000	.2164 ± .0000
	ProSVM-MU	.2079 ± .0000	.1628 ± .0000	.1338 ± .0000	.1234 ± .0000
	ProSVM-MD	**.1314 ± .0000**	**.1237 ± .0000**	**.1206 ± .0000**	**.1177 ± .0000**
	BSVM	.2883 ± .0009	.2412 ± .0007	.2286 ± .0002	.2173 ± .0000
	Maxide	.2595 ± .0000	.2523 ± .0000	.2464 ± .0000	.2376 ± .0000
YahooEntertainment	ProSVM-MI	.2228 ± .0001	.2056 ± .0000	.1978 ± .0000	.1924 ± .0000
	ProSVM-MU	.2112 ± .0001	.1796 ± .0000	.1508 ± .0000	.1367 ± .0000
	ProSVM-MD	**.1476 ± .0000**	**.1344 ± .0000**	**.1296 ± .0000**	**.1271 ± .0000**
	BSVM	.2946 ± .0004	.2592 ± .0003	.2367 ± .0003	.2193 ± .0001
	Maxide	.2314 ± .0002	.2321 ± .0001	.2278 ± .0001	.2205 ± .0000
YahooHealth	ProSVM-MI	.1811 ± .0001	.1663 ± .0001	.1584 ± .0001	.1527 ± .0000
	ProSVM-MU	.1548 ± .0001	.1210 ± .0001	.1057 ± .0001	.0991 ± .0000
	ProSVM-MD	**.1102 ± .0001**	**.1044 ± .0001**	**.0976 ± .0001**	**.0950 ± .0000**
	BSVM	.2730 ± .0001	.2587 ± .0000	.2507 ± .0000	.2450 ± .0000
	Maxide	.2815 ± .0001	.2715 ± .0000	.2630 ± .0000	.2563 ± .0000
YahooRecreation	ProSVM-MI	.2610 ± .0001	.2448 ± .0000	.2339 ± .0001	.2301 ± .0000
	ProSVM-MU	.2546 ± .0001	.2225 ± .0000	.1917 ± .0001	.1739 ± .0001
	ProSVM-MD	**.1893 ± .0000**	**.1691 ± .0000**	**.1594 ± .0001**	**.1579 ± .0000**
	BSVM	.3039 ± .0004	.2583 ± .0003	.2381 ± .0004	.2232 ± .0003
	Maxide	.2096 ± .0001	.2190 ± .0001	.2138 ± .0001	.2088 ± .0001
YahooReference	ProSVM-MI	.1842 ± .0001	.1756 ± .0001	.1695 ± .0001	.1655 ± .0000
	ProSVM-MU	.1658 ± .0001	.1417 ± .0001	.1260 ± .0001	.1135 ± .0001
	ProSVM-MD	**.1210 ± .0001**	**.1082 ± .0001**	**.1040 ± .0001**	**.1018 ± .0001**
	BSVM	.3182 ± .0003	.2871 ± .0004	.2676 ± .0002	.2533 ± .0001
	Maxide	.2686 ± .0001	.2646 ± .0000	.2543 ± .0001	.2448 ± .0001
YahooScience	ProSVM-MI	.2589 ± .0001	.2462 ± .0001	.2386 ± .0001	.2321 ± .0001
	ProSVM-MU	.2403 ± .0001	.2032 ± .0001	.1773 ± .0001	.1621 ± .0001
	ProSVM-MD	**.1735 ± .0002**	**.1620 ± .0001**	**.1535 ± .0001**	**.1488 ± .0001**
	BSVM	.2215 ± .0003	.1972 ± .0001	.1884 ± .0002	.1797 ± .0001
	Maxide	.2153 ± .0001	.2171 ± .0001	.2087 ± .0001	.2060 ± .0000
YahooSocial	ProSVM-MI	.1606 ± .0001	.1537 ± .0000	.1473 ± .0000	.1441 ± .0000
	ProSVM-MU	.1373 ± .0001	.1124 ± .0001	.0966 ± .0000	.0922 ± .0000
	ProSVM-MD	**.1016 ± .0000**	**.0923 ± .0000**	**.0895 ± .0000**	**.0874 ± .0000**
	BSVM	.2907 ± .0002	.2724 ± .0002	.2642 ± .0002	.2539 ± .0002
	Maxide	.2882 ± .0000	.2825 ± .0001	.2747 ± .0001	.2620 ± .0000
YahooSociety	ProSVM-MI	.2453 ± .0001	.2342 ± .0000	.2265 ± .0000	.2211 ± .0000
	ProSVM-MU	.2190 ± .0000	.1945 ± .0000	.1784 ± .0000	.1669 ± .0000
	ProSVM-MD	**.1736 ± .0000**	**.1641 ± .0000**	**.1596 ± .0000**	**.1567 ± .0000**

"Algo." specifies the name of the algorithms. $\omega\%$ represents the percentage of observed label assignments in training instances. The best result and its comparable ones (pairwise single-tailed t-tests at 95 percent confidence level) are bold.

TABLE 10
Results on Large Multi-Label Datasets with Partial Labels Compared with More Recent Proposed Methods, Measured by Pro Loss

Data Set	P-S-D	P-S-U	P-S-I	safeML	GLOCAL	genEML
Bibtex	**0.2545**	0.2708	0.2751	0.3308	0.3464	NaN
Delicious	**0.2878**	0.3604	0.3379	NaN	0.3689	0.4565

The best results are bold. (P-S-D: ProSVM-MD; P-S-U: ProSVM-MU; and P-S-I: ProSVM-MI).

Loss. For our proposed PRO Loss incorporating pairwise comparison between relevant labels and irrelevant labels, it is not surprised to see that it achieves best performance on Ranking Loss. The F1 measure is recognized as suitable for class-imbalanced data. Since PRO Loss weights different label pairs, it is under expectation that it performs good when facing data whose number of relevant labels is much smaller.

6.4 Time Cost and Parallel Computing

Table 7 shows the training time (in seconds) . As can be seen, the time efficiencies of ProSVMs are comparable to most methods. Specifically, PD-Sparse and PfastreXML are the fastest since they are designed to solve extreme multi-label learning problem. Our proposal is much faster than PCn, PC0, and GMLC. ProSVM-A performs slightly faster than ProSVM, especially on larger data such as Enron. The time efficiency of ProSVM can be further improved using parallel computing in Fig. 4. Here each point is the relative time efficiency compared to the time efficiency using only one core, that is, the paralleled training time in seconds are divided by that on single core to make different datasets' results comparable in one figure. As can be seen, the time cost of ProSVM can be reduced by parallelization.

7 Experiments with Partial Labels

In this section, we compare our proposed ProSVM-P with state-of-the-art methods on multi-label learning with partial labels problem. We conduct experiments on the same datasets used in Section 6. On 20 relatively small data sets, to simulate partial labels, we adopt the same method as [14], i.e., first sampling 10 percent of the instances as test data, and for the remaining 90 percent, making $\{10\%, 20\%, 30\%, 40\%\}$ of the annotations observed and all others missing. We repeat the algorithm ten times and present the average results. On 2 large data sets, we only make 10 percent of the training annotations observe. For our ProSVM-P, we test three versions of it, i.e., ProSVM-PD, ProSVM-PI and ProSVM-PU, which are different in estimating $\mathbf{C}_P$. For regularization parameter λ, we conduct 10-fold cross validation and pick the best λ from $2^{\{-10,-8,\ldots,8,10\}}$ on small data and set it 1 on large data.

We compare ProSVM-P on small data sets with four classical state-of-the-art algorithms. The first is the BSVM method [8]. We train one model on each label using all the observed annotations as training information. As in Section 6, we use the LibSVM [55] with linear kernel as the base classifier and the regularization parameter is tuned in the same way as ProSVM-P. We also compare our proposal with two MC methods [15] called MC-b and MC-1, depending on how they treat the bias term. We further compare ProSVM-P with Maxide [14], a matrix completion algorithm using features and label correlations as side information. For MC and Maxide methods, we adopt the recommended parameter tuning strategy by authors. We then compare ProSVM-P on two large data sets with more recently proposed methods which can deal with partial labels, including GLOCAL [24], genEML [13] and safeML [17]. For GLOCAL and genEML, the parameters are selected in the same way as Section 6. For safeML, we try different parameters and select the one with the best performance.

The Pro Loss results on smallest datasets are shown in Table 8. For these datasets, the MC methods can give results within 24 hours, thus we compare ProSVM-P with four methods. From the results, we can see that ProSVM-PD always works the best, followed by ProSVM-PU. We further show the results on the remaining small datasets in Table 9 where MC methods are not able to give any results within 24 hours. Thus we compare our proposal with the other two methods. We can see that all the three ProSVM-P methods get the best three on all datasets. The Pro Loss results on two large datasets are shown in Table 10. We can see that on data sets with only 10 percent observed partial labels, our proposed algorithms can still perform the best. Specially, the results of ProSVM-U and ProSVM-I without knowing the number of relevant labels can still perform better than the recently proposed baselines.

8 Conclusion

This paper extended our preliminary research [19]. In this paper, we studied a new multi-label problem that in practice the user usually concerns about the prediction on labels as well as the ranking of relevant labels while the annotation information can be partial. To address our problem, we presented a new multi-label criterion, i.e., PRO Loss, and proposed the corresponding ProSVM algorithms. ProSVM was further extended to handle the partial labels problem. Experiments exhibited encouraging performance of our proposal. We will consider extending our proposal to the application of recommendation systems in future work.

Acknowledgments

This research was supported by the National Key R&D Program of China (2018YFB1004300) and the National Science Foundation of China (61751306, 61772262).

References

[1] Z. Zhou and M. Zhang, "Multi-label learning," in *Encyclopedia of Machine Learning and Data Mining*. Berlin, Germany: Springer, 2017, pp. 875–881.
[2] S. Burkhardt and S. Kramer, "Online multi-label dependency topic models for text classification," *Mach. Learn.*, vol. 107, no. 5, pp. 859–886, 2018.
[3] H. Wang, L. Yan, H. Huang, and C. H. Q. Ding, "From protein sequence to protein function via multi-label linear discriminant analysis," *IEEE/ACM Trans. Comput. Biol. Bioinf.*, vol. 14, no. 3, pp. 503–513, May/Jun. 2017.
[4] L. Song, J. Liu, B. Qian, M. Sun, K. Yang, M. Sun, and S. Abbas, "A deep multi-modal CNN for multi-instance multi-label image classification," *IEEE Trans. Image Process.*, vol. 27, no. 12, pp. 6025–6038, Dec. 2018.

[5] W. Cheng, E. Hüllermeier, W. Waegeman, and V. Welker, "Label ranking with partial abstention based on thresholded probabilistic models," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2012, pp. 2501–2509.

[6] H. Jain, Y. Prabhu, and M. Varma, "Extreme multi-label loss functions for recommendation, tagging, ranking & other missing label applications," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 935–944.

[7] Y. Prabhu, A. Kag, S. Gopinath, K. Dahiya, S. Harsola, R. Agrawal, and M. Varma, "Extreme multi-label learning with label features for warm-start tagging, ranking & recommendation," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2018, pp. 441–449.

[8] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown, "Learning multi-label scene classification," *Pattern Recognit.*, vol. 37, no. 9, pp. 1757–1771, 2004.

[9] R. Babbar and B. Schölkopf, "DISMEC: Distributed sparse machines for extreme multi-label classification," in *Proc. ACM Int. Conf. Web Search Data Mining*, 2017, pp. 721–729.

[10] S. Decubber, T. Mortier, K. Dembczynski, and W. Waegeman, "Deep F-measure maximization in multi-label classification: A comparative study," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2018, pp. 290–305.

[11] A. Elisseeff and J. Weston, "A kernel method for multi-labeled classification," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2001, pp. 681–687.

[12] J. Nam, E. Loza Mencía, H. J. Kim, and J. Fürnkranz, "Maximizing subset accuracy with recurrent neural networks in multi-label classification," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5413–5423.

[13] V. Jain, N. Modhe, and P. Rai, "Scalable generative models for multi-label learning with missing labels," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1636–1644.

[14] M. Xu, R. Jin, and Z.-H. Zhou, "Speedup matrix completion with side information: Application to multi-label learning," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2013, pp. 2301–2309.

[15] A. B. Goldberg, X. Zhu, B. Recht, J.-M. Xu, and R. D. Nowak, "Transduction with matrix completion: Three birds with one stone," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2010, pp. 757–765.

[16] B. Wu, F. Jia, W. Liu, B. Ghanem, and S. Lyu, "Multi-label learning with missing labels using mixed dependency graphs," *Int. J. Comput. Vis.*, vol. 126, no. 8, pp. 875–896, 2018.

[17] T. Wei, L. Guo, Y. Li, and W. Gao, "Learning safe multi-label prediction for weakly labeled data," *Mach. Learn.*, vol. 107, no. 4, pp. 703–725, 2018.

[18] X. Zhu, "Semi-supervised learning," in *Encyclopedia of Machine Learning*. Berlin, Germany: Springer, 2010, pp. 892–897.

[19] M. Xu, Y.-F. Li, and Z.-H. Zhou, "Multi-label learning with PRO loss," in *Proc. AAAI Conf. Artif. Intell.*, 2013, pp. 998–1004.

[20] R. S. Cabral, F. D. la Torre, J. P. Costeira, and A. Bernardino, "Matrix completion for weakly-supervised multi-label image classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 1, pp. 121–135, Jan. 2015.

[21] W. Liu, I. W. Tsang, and K. Müller, "An easy-to-hard learning paradigm for multiple classes and multiple labels," *J. Mach. Learn. Res.*, vol. 18, pp. 94:1–94:38, 2017.

[22] M. Zhang, Y. Li, X. Liu, and X. Geng, "Binary relevance for multi-label learning: An overview," *Frontiers Comput. Sci.*, vol. 12, no. 2, pp. 191–202, 2018.

[23] W. Bi and J. T. Kwok, "Bayes-optimal hierarchical multilabel classification," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 2907–2918, Nov. 2015.

[24] Y. Zhu, J. T. Kwok, and Z. Zhou, "Multi-label learning with global and local label correlation," *IEEE Trans. Knowl. Data Eng.*, vol. 30, no. 6, pp. 1081–1094, Jun. 2018.

[25] K. Dembczynski, W. Waegeman, W. Cheng, and E. Hüllermeier, "On label dependence and loss minimization in multi-label classification," *Mach. Learn.*, vol. 88, no. 1/2, pp. 5–45, 2012.

[26] M. Gasse, A. Aussem, and H. Elghazel, "On the optimality of multi-label classification under subset zero-one loss for distributions satisfying the composition property," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 2531–2539.

[27] A. Kanehira and T. Harada, "Multi-label ranking from positive and unlabeled data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 5138–5146.

[28] Y. Li, Y. Song, and J. Luo, "Improving pairwise ranking for multi-label image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1837–1845.

[29] O. Koyejo, N. Natarajan, P. Ravikumar, and I. S. Dhillon, "Consistent multilabel classification," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 3321–3329.

[30] J. Fürnkranz, E. Hüllermeier, E. Loza Mencía, and K. Brinker, "Multilabel classification via calibrated label ranking," *Mach. Learn.*, vol. 73, no. 2, pp. 133–153, 2008.

[31] W. Cheng, K. Dembczynski, and E. Hüllermeier, "Graded multilabel classification: The ordinal case," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 223–230.

[32] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *Nat. Sci. Rev.*, vol. 5, no. 1, pp. 44–53, 2017.

[33] W. Bi and J. T. Kwok, "Multilabel classification with label correlations and missing labels," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 1680–1686.

[34] X. Li, F. Zhao, and Y. Guo, "Conditional restricted Boltzmann machines for multi-label learning with incomplete labels," in *Proc. Int. Conf. Artif. Intell. Statist.*, 2015, pp. 635–643.

[35] H.-F. Yu, P. Jain, P. Kar, and I. S. Dhillon, "Large-scale multi-label learning with missing labels," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. I-593–I-601.

[36] L. Jing, L. Yang, J. Yu, and M. K. Ng, "Semi-supervised low-rank mapping learning for multi-label classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 1483–1491.

[37] X. V. Lin, S. Singh, L. He, B. Taskar, and L. Zettlemoyer, "Multi-label learning with posterior regularization," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2014, http://www.cs.cmu.edu/~apparikh/nips2014ml-nlp/posters.html

[38] X. Wu and Z. Zhou, "A unified view of multi-label performance measures," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3780–3788.

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

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

[41] C. Hsieh, Y. Lin, and H. Lin, "A deep model with local surrogate loss for general cost-sensitive multi-label learning," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 3239–3246.

[42] K. Bhatia, H. Jain, P. Kar, M. Varma, and P. Jain, "Sparse local embeddings for extreme multi-label classification," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2015, pp. 730–738.

[43] Y. Prabhu and M. Varma, "FastXML: A fast, accurate and stable tree-classifier for extreme multi-label learning," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2014, pp. 263–272.

[44] I. E. Yen, X. Huang, P. Ravikumar, K. Zhong, and I. S. Dhillon, "PD-sparse : A primal and dual sparse approach to extreme multiclass and multilabel classification," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 3069–3077.

[45] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[46] C. Leng, Z. Dou, H. Li, S. Zhu, and R. Jin, "Extremely low bit neural network: Squeeze the last bit out with ADMM," in *Proc. AAAI Conf. Artif. Intell.*, pp. 3466–3473, 2018.

[47] X. Zhang, M. M. Khalili, and M. Liu, "Improving the privacy and accuracy of ADMM-based distributed algorithms," in *Proc. Int. Conf. Mach. Learn.*, pp. 5791–5800, 2018.

[48] P. A. Forero, A. Cano, and G. B. Giannakis, "Consensus-based distributed support vector machines," *J. Mach. Learn. Res.*, vol. 11, pp. 1663–1707, 2010.

[49] W. Kotlowski, K. Dembczynski, and E. Hüllermeier, "Bipartite ranking through minimization of univariate loss," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 1113–1120.

[50] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, 2008.

[51] B. He and X. Yuan, "On the O(1/n) convergence rate of the Douglas-Rachford alternating direction method," *SIAM J. Numerical Anal.*, vol. 50, no. 2, pp. 700–709, 2012.

[52] Y. Wang, P. Lin, and Y. Hong, "Distributed regression estimation with incomplete data in multi-agent networks," *Sci. China Inf. Sci.*, vol. 61, no. 9, pp. 092 202:1–092 202:14, 2018.

[53] E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker, "Label ranking by learning pairwise preferences," *Artif. Intell.*, vol. 172, no. 16/17, pp. 1897–1916, 2008.

[54] S. Ubaru and A. Mazumdar, "Multilabel classification with group testing and codes," in *Proc. Int. Conf. Mach. Learn.*, pp. 3492–3501, 2017.

[55] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011, Art. no. 27.

[56] M.-L. Zhang and Z.-H. Zhou, "Multilabel neural networks with applications to functional genomics and text categorization," *IEEE Trans. Knowl. Data Eng.*, vol. 18, no. 10, pp. 1338–1351, Oct. 2006.

[57] M. L. Zhang and Z.-H. Zhou, "Multi-label learning by instance differentiation," in *Proc. AAAI Conf. Artif. Intell.*, 2007, pp. 669–674.

[58] M.-L. Zhang, J. M. Peña, and V. Robles, "Feature selection for multi-label Naive Bayes classification," *Inf. Sci.*, vol. 179, no. 19, pp. 3218–3229, 2009.

[59] H. Li, M. Wang, and X.-S. Hua, "MSRA-MM 2.0: A large-scale web multimedia dataset," in *Proc. IEEE Int. Conf. Data Mining Workshop*, 2009, pp. 164–169.

[60] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, 2006.

[61] J. Huang, G. Li, Q. Huang, and X. Wu, "Learning label-specific features and class-dependent labels for multi-label classification," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 12, pp. 3309–3323, Dec. 2016.

[62] Q. Wu, M. Tan, H. Song, J. Chen, and M. K. Ng, "ML-FOREST: A multi-label tree ensemble method for multi-label classification," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 10, pp. 2665–2680, Oct. 2016.

Miao Xu received the BSc and PhD degrees in computer science from Nanjing University, China, in 2009 and 2017, respectively. Currently, she is a postdoctoral researcher with the RIKEN Center for Advanced Intelligence Project. Her main research interests include machine learning and data mining. She won the IBM PhD Fellowship Award in 2015.

Yu-Feng Li received the BSc and PhD degrees in computer science from Nanjing University, China, in 2006 and 2013, respectively. He joined Nanjing University, in 2013, and is currently an associate professor of the National Key Laboratory for Novel Software Technology. He is a member of the LAMDA Group. His research interests are mainly in machine learning. Particularly, he is interested in semi-supervised learning, statistical learning, and optimization. He has published more than 30 papers in top-tier journal and conferences such as the *Journal of Machine Learning Research*, the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *Artificial Intelligence Journal*, ICML, NIPS, AAAI, etc. He is/has served as a senior program committee member of top-tier AI conferences such as IJCAI15/17/19, AAAI19, and an editorial board member of the *Machine Learning Journal* special issues. He has received Outstanding Doctoral Dissertation Award from China Computer Federation (CCF), Outstanding Doctoral Dissertation Award from Jiangsu Province, and Microsoft Fellowship Award.

Zhi-Hua Zhou (S'00-M'01-SM'06-F'13) received the BSc, MSc, and PhD degrees in computer science from Nanjing University, China, in 1996, 1998, and 2000, respectively, all with the highest honors. He joined the Department of Computer Science & Technology, Nanjing University, as an assistant professor, in 2001, and is currently professor, head of the Department of Computer Science and Technology, and dean of the School of Artificial Intelligence. He is also the founding director of the LAMDA Group. His research interests are mainly in artificial intelligence, machine learning, and data mining. He has authored the books *Ensemble Methods: Foundations and Algorithms* and *Machine Learning* (in Chinese), and published more than 150 papers in top-tier international journals or conference proceedings. He has received various awards/honors including the National Natural Science Award of China, the PAKDD Distinguished Contribution Award, the IEEE ICDM Outstanding Service Award, the Microsoft Professorship Award, etc. He also holds 24 patents. He is the editor-in-chief of the *Frontiers of Computer Science*, associate editor-in-chief of *Science China Information Sciences*, action or associate editor of the *Machine Learning*, the *IEEE Transactions on Pattern Analysis and Machine Intelligence*, the *ACM Transactions on Knowledge Discovery from Data*, etc. He served as associate editor-in-chief of the *Chinese Science Bulletin* (2008-2014), associate editor of the *IEEE Transactions on Knowledge and Data Engineering* (2008-2012), the *IEEE Transactions on Neural Networks and Learning Systems* (2014-2017), the *ACM Transactions on Intelligent Systems and Technology* (2009-2017), the *Neural Networks* (2014-2016), etc. He founded ACML (Asian Conference on Machine Learning), served as an advisory committee member for IJCAI (2015-2016), steering committee member for ICDM, PAKDD, and PRICAI, and chair of various conferences such as general co-chair of ICDM 2016 and PAKDD 2014, program co-chair of AAAI 2019 and SDM 2013, and area chair of NIPS, ICML, AAAI, IJCAI, KDD, etc. He is/was the chair of the IEEE CIS Data Mining Technical Committee (2015-2016), the chair of the CCF-AI (2012-), and the chair of the CAAI Machine Learning Technical Committee (2006-2015). He is a foreign member of the Academy of Europe, and a fellow of the ACM, AAAI, AAAS, IEEE, IAPR, IET/IEE, CCF, and CAAI.