Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence

Multi-Label Learning with PRO Loss

Miao Xu Yu-Feng Li Zhi-Hua Zhou*
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China
{xum,liyf,zhouzh}@lamda.nju.edu.cn

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 the *relevant* labels for an object, whereas the rankings of *irrelevant* labels are not important. 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 on all labels as well as the rankings 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. Experiments show that our proposals are not only superior on PRO LOSS, but also highly competitive on existing evaluation criteria.

Introduction

In real applications, one object may be associated with multiple labels simultaneously, and such problems are realized by multi-label learning (Tsoumakas, Katakis, and Vlahavas 2010). During the past decade, many multi-label methods have been developed and found useful in diverse applications (Schapire and Singer 2000; Elisseeff and Weston 2002; Boutell et al. 2004; Kazawa et al. 2005; Yu, Yu, and Tresp 2005; Barutcuoglu, Schapire, and Troyanskaya 2006; Qi et al. 2007).

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 label is relevant and which is irrelevant. In many applications, however, in addition to label prediction, there is often another requirement, i.e., to get good rankings of the predicted relevant labels. Consider a simple example.

*This research was supported by NSFC (61073097), 973 Program (2010CB327903), JiangsuSF (BK2011566) and Huawei Fund (YBCB2012085).

Figure 1: Ordered relevant labels of images. Left: {*cattle*, *mountain*, *road*}, Right: {*mountain*, *road*, *cattle*}.

Both images in Figure 1 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 first one and {*mountain*, *road*, *cattle*} for the second one. It is noteworthy that although the rankings of relevant labels are important, the rankings of irrelevant labels, which does not occur within any image, are not useful.

In practice, the relevant label ordering information can be obtained, for example in crowdsourcing applications, by counting the supporters for each label. Such a learning problem, however, cannot be addressed by typical multi-label learning methods that focus on label prediction because they generally ignore the rankings of relevant labels. For example, the BR approach (Boutell et al. 2004) simply trains a binary model for each label; RankSVM (Elisseeff and Weston 2002; Fürnkranz et al. 2008) focuses on distinguishing the relevant labels from irrelevant ones; BoosTexter (Schapire and Singer 2000) and ML-kNN (Zhang and Zhou 2007a) focus on improving generalization of label predictions by exploiting label correlations. It is also notable that our concerned problem cannot be addressed by typical label ranking approaches (Dekel, Manning, and Singer 2003; Gärtner and Vembu 2010; Hüllermeier et al. 2008; Shalev-Shwartz and Singer 2006), which focus on learning a mapping from instances to rankings over a predefined set of labels. To adapt them to our concerned problem there needs a non-trivial process to address the challenging issue of selecting the "cut-point" in the label ordering for deciding the relevant ones. The PC (Pairwise Comparison with calibrated label ranking) method (Fürnkranz et al. 2008) considers a combination of multi-label learning and label ranking by

creating an additional calibrated label. However, it concerns about either "multi-label learning" or "label ranking" without recognizing that only the rankings of relevant labels are crucial. Moreover, PC treats the label pairs independently and may produce inconsistent results; for example, given three labels A, B and C, it may predict A>B, B>C but C>A. Recently, Cheng et al. 2010 propose a related label ranking method GMLC which assumes that the labels of an object are categorized into multiple degrees of relevances; in contrast, we do not assume the existence of such information.

The infeasibility of these existing methods on our concerned problem might owe to the fact that they were designed to optimize the state-of-the-art performance criteria. For example, BR was tailored for HAMMING LOSS; RankSVM was designed for RANKING LOSS; AdaBoost.MH and Adaboost.MR (Schapire and Singer 2000), two implementations of BoosTexter, were designed to optimize HAMMING LOSS and RANKING LOSS, respectively. As we will discuss comprehensively in the next section, however, none of the state-of-the-art criteria is able to express the requirement of our concerned problem precisely. Therefore, to address our problem, new criterion as well as new algorithms are needed.

In this paper, we present the PRO LOSS (Prediction and Relevance Ordering Loss), a new multi-label criterion that concerns the label predictions as well as the rankings 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. Experiments show that our proposals are not only superior to state-of-the-art approaches on PRO LOSS, but also highly competitive on existing evaluation criteria.

The rest of the paper is organized as follows. We first revisit existing criteria. Then we present PRO LOSS and ProSVMs, followed by experiments and conclusion.

Existing Criteria Revisited

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 one ranked relevant label set $R_i \subseteq L$ and corresponding irrelevant label set $\overline{R}_i = L - R_i$, on which the rankings are not concerned.

Existing multi-label learning algorithms typically learn a function $\mathbf{g}(\mathbf{x}_i) = [g_1(\mathbf{x}_i), \cdots, g_T(\mathbf{x}_i)]$ that will assign a 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 further determine a threshold, denoted by $g_\Theta(\mathbf{x}_i)$. Those labels with scores larger than the threshold will be regarded as relevant ones, otherwise irrelevant ones. Here $g_\Theta(\mathbf{x}_i)$ can be simply set to 0; it can also be set more accurately by learning from data (Fürnkranz et al. 2008). 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 (Schapire and Singer 2000; Elisseeff and Weston 2002; Fürnkranz et al. 2008)

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

  Here $\triangle$ stands for the symmetric difference between two sets. Obviously, the HAMMING LOSS ignores the fact that different relevant labels may have different priorities.

- RANKING LOSS (Schapire and Singer 2000; Elisseeff and Weston 2002; Yu, Yu, and Tresp 2005)

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

  Here δ is the indicator function. RANKING LOSS concerns the relative rankings in each relevant-irrelevant label pair. However, it does not consider the rankings of relevant labels.

- ONE-ERROR (Schapire and Singer 2000; Elisseeff and Weston 2002; Zhang and Zhou 2007a)

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

  ONE-ERROR considers the top predicted relevant label only and thus neglecting all the other relevant labels.

- AVERAGE PRECISION (Schapire and Singer 2000; Elisseeff and Weston 2002; Zhang and Zhou 2007a)

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

  AVERAGE PRECISION concerns the wrong prediction of irrelevant labels only if they are ranked above all relevant labels.

- COVERAGE (Schapire and Singer 2000)

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

  COVERAGE concerns the worst predicted relevant label only and thus neglecting the other relevant labels.

- SUBSET ACCURACY (Dembczynski et al. 2010)

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

  SUBSET ACCURACY does not consider label ordering.

- F1 (Godbole and Sarawagi 2004)

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

  Alternative definitions include MACRO-F1 and MICRO-F1 (Yang 1999) which are averaged over labels instead of instances. F1 does not take any rankings of relevant labels into account.

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

PRO LOSS

We first introduce some notations. Given an instance $\mathbf{x}$ and its relevant label set R, we denote by $\prec_{\mathbf{x}}(a)$ the set of indices of labels that are less relevant than l_a. We separate the labels into three groups, i.e., relevant, threshold and irrelevant, and denote by $\mathcal{B}(a)$ the set of indices of labels that are in the same subgroup of l_a. For example, 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, then we have $\prec_{\mathbf{x}}(1) = \{2, \Theta, 3, 4\}$, $\prec_{\mathbf{x}}(2) = \{\Theta, 3, 4\}$, $\prec_{\mathbf{x}}(\Theta) = \{3, 4\}$, $\prec_{\mathbf{x}}(3) = \prec_{\mathbf{x}}(4) = \emptyset$, $\mathcal{B}(1) = \mathcal{B}(2) = \{1, 2\}$, $\mathcal{B}(3) = \mathcal{B}(4) = \{3, 4\}$ and $\mathcal{B}(\Theta) = \{\Theta\}$.

We then define the PRO LOSS for an instance $\mathbf{x}$ as:

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

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.

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 ordering of any two irrelevant labels is not valued in Eq. 1. Hence, PRO LOSS considers an accurate label prediction as well as a correct relevance ordering.

To balance these label pairs to avoid dominated terms, 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 $|\overline{R}|$, which can be written in a general form as

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

This leads to our PRO LOSS.

ProSVMs

Note that $\ell_{t,s}$, a modified 0-1 loss, is non-convex and difficult to optimize, we consider optimizing a large margin surrogate convex loss (Vapnik 1998) as follows:

$$\min_{\mathbf{g}} \quad \lambda \sum_{i=1}^{n} \widehat{\mathcal{L}}(\mathbf{x}_i, R_i, \prec, \mathbf{g}) + \Omega(\mathbf{g}), \quad (2)$$

where $\Omega(\mathbf{g})$ is a regularizer for $\mathbf{g}$, $\widehat{\mathcal{L}}(\mathbf{x}_i, R_i, \prec, \mathbf{g}) = \sum_{l_t \in R_i \cup \{\Theta\}} \sum_{s \in \prec_{\mathbf{x}_i}(t)} \frac{1}{4h_{s,t}} (1 + g_s(\mathbf{x}_i) - g_t(\mathbf{x}_i))_+$ 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 $\Omega(\mathbf{g}) = \frac{1}{2} \sum_{t \in \{1,\ldots,T\} \cup \{\Theta\}} \|\mathbf{w}_t\|^2$. Let $\bar{\mathbf{w}} \triangleq [\mathbf{w}_1; \ldots; \mathbf{w}_T; \mathbf{w}_\Theta]$ and D be the training set. Note that $\widehat{\mathcal{L}}(\mathbf{x}_i, R_i, \prec, \mathbf{g})$ is no more than a sum of hinge losses, Eq. 2 is then cast as an SVM-type problem in the following general form:

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

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 for instances. Due to space limitation, they will be presented in a longer version.

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

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

An Efficient Algorithm

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

ADMM (Bertsekas and Tsitsiklis 1989) is a simple and efficient approach for large scale optimization. Its basic idea is to take the *decomposition-coordinate* procedure such that the solution of subproblems can be coordinated to find the solution to the original problem. Since subproblems can usually be efficiently solved, ADMM is capable of approximating the solution of large scale problems via addressing small subproblems sequentially. Moreover, ADMM is easy to parallelize and therefore, does not suffer the memory capacity problem. Recently, ADMM has been found effective on many machine learning problems (Boyd et al. 2011; Forero, Cano, and Giannakis 2010).

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

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

By introducing the surrogate augmented lagrangian function (Forero, Cano, and Giannakis 2010) for Eq.5, we have,

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

[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 $1/2$ by average.

Algorithm 1 ProSVM

1: Decompose data set D into Z disjoint subsets, i.e., $D^1, \ldots, D^Z$. Set $k = 0$.
2: Initialize $\{\bar{\mathbf{w}}_0^0, \bar{\mathbf{w}}_0^1, \cdots, \bar{\mathbf{w}}_0^Z, \alpha_0^1, \cdots, \alpha_0^Z\}$ as zeros.
3: **while** not converge **do**
4: Set $k = k + 1$ and update $\{\bar{\mathbf{w}}_k^0, \{\bar{\mathbf{w}}_k^z, \alpha_k^z\}_{z=1}^Z\}$ as:

$$\{\bar{\mathbf{w}}_k^z\}_{z=1}^Z = \arg\min_{\bar{\mathbf{w}}^1, \cdots, \bar{\mathbf{w}}^Z} \mathbb{L}(\bar{\mathbf{w}}_{k-1}^0, \{\bar{\mathbf{w}}^z, \alpha_{k-1}^z\}_{z=1}^Z, \eta) \quad (6)$$

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

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

5: **end while**
6: **Output** Final $\bar{\mathbf{w}}^0$

where α^z's and η are the lagrange multipliers. $\mathbb{L}$ is then solved in an iterative manner, i.e., updating the solutions of $\{\bar{\mathbf{w}}^1, \cdots, \bar{\mathbf{w}}^Z\}$, $\{\bar{\mathbf{w}}^0\}$ and $\{\alpha^z\}_{z=1}^Z$ separately and iteratively until convergence. Detailed updating processes are shown in Algorithm 1. According to the theoretical finding in (He and Yuan 2012), it is not hard to show that our algorithm will converge to a global optimal solution in the convergence rate of $O(1/K)$ where K is the number of iterations. Note that although theoretically $O(1/K)$ is not a fast convergence rate, in practice, optimal solution is usually not necessary (i.e., a good approximate solution is already sufficient to obtain a satisfactory performance) (Boyd et al. 2011). In our experiment, the maximal iteration is simply set to 100 and empirical results validate our effectiveness.

Note that the key to have efficient ProSVMs is to efficiently solve Eqs. 6 and 7. As for Eq. 6, it is equivalent to solve the following Z independent smaller subproblems.

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

which is a convex quadratic programming (QP) problem. Furthermore, note 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 like LIBLINEAR (Fan et al. 2008). 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.

Reduce the Number of Comparisons

There are $O(T|R|)$ constraints in total for each instance in Eq. 2, where $|R|$ typically scales to $O(T)$. Thus, the number of constraints then scales to $O(T^2)$ which is too many 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. According to the work in (Kotlowski, Dembczynski, and Huellermeier 2011), we get the following theorem.

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:*

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

Theorem 1 shows that the relevant-irrelevant label pairs can be approximated by the relevant-threshold and irrelevant-threshold pairs which both scale to $O(T)$ only. Next we consider simplifying the number of comparisons between relevant labels. Our basic idea is to approximate full pairs of comparisons between relevant labels with a chain of comparisons between a relevant label and its immediate follower, which also scales to $O(T)$.

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

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

According to Theorems 1 and 2, one can approximate the objective function in Eq. 2 with an upper bound, i.e.,

$$\sum_{l_i \in R} \frac{\ell_{i,\Theta}}{2|\mathcal{B}(i)|} + \sum_{l_j \in \overline{R}} \frac{\ell_{\Theta,j}}{2|\mathcal{B}(j)|} + \sum_{i=1}^{|R|-1} \frac{(i(|R| - i)\ell_{r_i, r_{i+1}})}{2|R|(|R| - 1)}, \quad (9)$$

in which the number of constraints only scales to $O(T)$. Note that Eq. 9 can be addressed via the same optimization techniques as Eq. 2. We refer to this new algorithm as ProSVM-A (ProSVM Approximation).

Experiments

Our proposals are compared with a number of state-of-the-art multi-label methods, including PC (Fürnkranz et al. 2008), RankSVM (Elisseeff and Weston 2002), BSVM (Boutell et al. 2004), ML-kNN (Zhang and Zhou 2007a) and BoosTexter (Schapire and Singer 2000). For PC, Perceptron is employed as the base learner following (Fürnkranz et al. 2008). Two implementations of PC, i.e., PCn and PC0, are considered. In PCn, Perceptron stops after n rounds while in PC0, it stops when no error occurs or reaching 10000 rounds. One simple approach to extend PC for our concerned problem is to incorporate rankings of relevant labels. We also compare with these variants of PC, namely PCnR and PC0R, respectively. Another simple baseline is to first predict the relevant labels, and then rank them. Here we use RankSVM (Elisseeff and Weston 2002) for prediction, and then employ Pairwise Comparison (Hüllermeier et al. 2008) for ranking. We call the resulted algorithm as RankSVM-R. Moreover, we also compare with GMLC (Cheng, Dembczyński, and Hüllermeier 2010) which considers multiple degrees of label relevances. To run GMLC, the number of relevance levels is fixed to be $\max_{i=1}^n(|R_i|+1)$, and the i-th relevant label is assigned to the i-th level while the irrelevant labels are assigned to the $(\max_{i=1}^n(|R_i|+1))$-th level. It is noteworthy that although most of the compared algorithms

Table 1: Results (mean±std) on MSRA-M with real ordering. The best result and its comparable ones (pairwise t-test at 95% confidence) are bolded. RSVM(-R) shorts for RankSVM(-R). BTX shorts for BoosTexter.

METHOD	PRO LOSS	METHOD	PRO LOSS
PROSVM	**.2562±.0114**	RSVM	.2992±.0144
PROSVM-A	**.2606±.0128**	RSVM-R	.2609±.0116
PCN	.3754±.0406	BSVM	.2913±.0070
PCNR	.3469±.0420	MLkNN	.3228±.0099
PC0	.3149±.0107	BTX	.2957±.0112
PC0R	.3040±.0090	GMLC	.3052±.0130

are not designed for relevant label ordering, they are able to perform label ordering by comparing the predicted scores on labels, from which we can calculate PRO LOSS for these algorithms.

The setups of our proposals and compared methods are as follows. For RankSVM, the regularization parameter is selected from $\{2^{-10}, 2^{-8}, ..., 2^{8}, 2^{10}\}$ by ten-fold cross validation. For BSVM, the SVM is implemented by LIBSVM (Chang and Lin 2011) package with parameters selected in the same way as RankSVM. For ML-kNN, we use the parameter setting recommended by (Zhang and Zhou 2007a). For BoosTexter, we use the version AdaBoost.MH (Schapire and Singer 2000). For ProSVMs, λ is chosen by ten-fold cross validation and 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.

Data with Real Label Ordering

It is notable that the problem of relevance ordering is relatively new, and the required multi-label data sets are not widely available yet. Here we provide the first real data MSRA-M with relevance ordering. Specifically, we use a subset of the widely-used MSRA data set (Li, Wang, and Hua 2009) which contains 1868 images. Each instance/image is represented by a 899 dimensional feature vector. There are 19 candidate labels. Each image belongs to 1 to 11 relevant ones. The ordering of relevant labels are manually provided by image curators. In our experiment, 10-CV is conducted and the average results are recorded. Results are shown in Table 1. As can be seen, ProSVMs perform significantly better than all other compared methods.

Data with Synthetic Label Ordering

Except for MSRA-M, the current public multi-label data sets do not contain label ordering information. To employ them, we automatically simulate the relevance ordering by running 3 state-of-the-art multi-label methods (Zhang and Zhou 2006; 2007b; Zhang, Peña, and Robles 2009) each predicts a real score for each label, and then obtain the ordering of relevant labels by sorting the aggregated real scores. By this approach, a broad range of 19 data sets which cover diverse domains, e.g., *music*, *biology*, *image* and *text*, are studied[2], The number of samples varies from 590 to 5,000, the number of dimensionality varies from 72 to 1,449 and the number of labels varies from 5 to 53. The results are shown in Table 2. As can be seen, ProSVMs perform superior to compared methods. In particular, ProSVM achieves the best result on 13 over 19 data sets while ProSVM-A achieves the best result on the rest 6 data sets.

Data without Label Ordering

Our next experiment is to study the performance of our proposals on existing criteria. Here our proposals are evaluated by neglecting the relevance ordering information. Specifically, a simpler loss function without considering the pairs of relevant labels is used for ProSVMs, and the optimization techniques employed in ProSVMs are applied to solve the new simpler objective. We call our new variants as ProSVM' and ProSVM-A'. Note that PCnR, PC0R and RankSVM-R could not be compared since they require for relevant label ordering information. For GMLC, two relevance levels, i.e., relevant and irrelevant, are used.

We plot the *robustness* of the criteria in Figure 2. The *robustness* was designed by (Zhou and Yu 2005); roughly speaking, given a data set, for a concerned criterion which is the smaller the better, the worst-performed algorithm is identified at first, and then the relative performance of all the algorithms is obtained by dividing their loss value by the worst one; the results of one algorithm are aggregated across all data sets, and the final aggregated value provides a good indication of the robustness of the algorithm. As can be seen, even without the relevance ordering information, our proposals still perform highly competitive to state-of-the-art multi-label methods on existing criteria.

Time Cost and Parallel Computing

Figure 3(a) shows the *robustness* of time cost of our proposals and compared methods. As can be seen, the time efficiencies of ProSVMs are comparable to most compared methods. The efficiency of using multi-core on representative eight data sets are illustrated in Figure 3(b). As can be seen, the time cost of ProSVM can be significantly reduced by parallel computing.

Conclusion

In this paper, we study a new multi-label problem that in practice the user usually concerns about the prediction on labels as well as the ordering among relevant labels. To address our problem, we present a new multi-label criterion, i.e., PRO LOSS, and propose the ProSVMs that optimize this new loss. Experiments exhibit encouraging performance of our proposals. The theoretical analysis of PRO LOSS will be studied in future.

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

Table 2: Comparison results on PRO LOSS for data with synthetic label ordering. Each entry presents the PRO LOSS; the best result on each data is bolded. RSVM(-R) is short for RankSVM(-R). MLk is short for MLkNN. Btx is short for BoosTexter. For IMAGE and SLASHDOT that have not provided training/testing splits, 10-CV is conducted and average performances are recorded. For others we use the provided training/testing splits . The last row presents the sum of ranks; the smaller the R-total, the better the overall performance.

DATA SET	PROSVM	PROSVM-A	PCN	PCNR	PC0	PC0R	RSVM	RSVM-R	BSVM	MLk	BTX	GMLC
EMOTIONS	**.1997**	.2090	.3557	.3509	.2821	.2641	.2159	.2110	.2164	.2210	.2397	.2255
ENRON	**.1497**	.1547	.3015	.3032	.3143	.3031	.1507	.1587	.2335	.2533	.2121	.3913
GENBASE	**.0023**	.0027	.2544	.2544	.0511	.0489	.0063	.0074	.0269	.0181	.0049	.0109
IMAGE	.1645	**.1638**	.2755	.2738	.2481	.2518	.2079	.2086	.1896	.1914	.1737	.2150
MEDICAL	**.0591**	.0599	.2769	.2769	.2038	.1998	.0940	.0935	.1296	.1647	.0838	.1510
SCENE	**.1031**	.1047	.2829	.2840	.2710	.2713	.1198	.1243	.1313	.1228	.1081	.1405
SLASHDOT	**.1158**	.1173	.2877	.2877	.2781	.2766	.1686	.1689	.2052	.2944	.1793	.3632
YAHOOARTS	.1500	**.1496**	.3176	.3179	.3062	.3060	.2288	.2307	.2276	.3067	.2474	.3887
YAHOOBUSINESS	**.0605**	.0621	.2673	.2673	.1713	.1713	.0837	.0849	.2725	.0921	.0912	.1207
YAHOOCOMPUTERS	**.0989**	.1045	.2861	.2864	.1599	.1599	.1918	.1918	.1185	.2073	.1852	.2776
YAHOOEDUCATION	.1113	**.1091**	.2951	.2939	.1830	.1828	.2123	.2129	.2176	.2479	.2264	.3292
YAHOOENTERTAINMENT	.1179	**.1178**	.2955	.2933	.1677	.1674	.1865	.1875	.2437	.2419	.2064	.3118
YAHOOHEALTH	**.0899**	.0944	.3045	.2961	.1553	.1547	.1474	.1504	.2126	.2044	.1619	.2944
YAHOORECREATION	**.1531**	.1536	.3026	.3018	.2800	.2803	.2249	.2256	.2365	.3045	.2438	.3714
YAHOOREFERENCE	.0931	**.0919**	.2779	.2779	.1480	.1485	.1565	.1566	.2491	.2296	.1783	.3135
YAHOOSCIENCE	**.1388**	.1489	.2985	.2988	.2154	.2157	.2288	.2294	.2021	.2628	.2480	.3448
YAHOOSOCIAL	**.0863**	.0889	.2853	.2856	.1630	.1626	.1356	.1369	.2598	.1648	.1542	.2752
YAHOOSOCIETY	.1517	**.1506**	.3114	.3111	.2654	.2632	.2199	.2199	.1741	.2280	.2308	.2993
YEAST	**.1854**	.1869	.3472	.3406	.4177	.4141	.1933	.2605	.2493	.2338	.2548	.2326
R-TOTAL	25	33	207	203	143	135	76	95	131	146	103	185

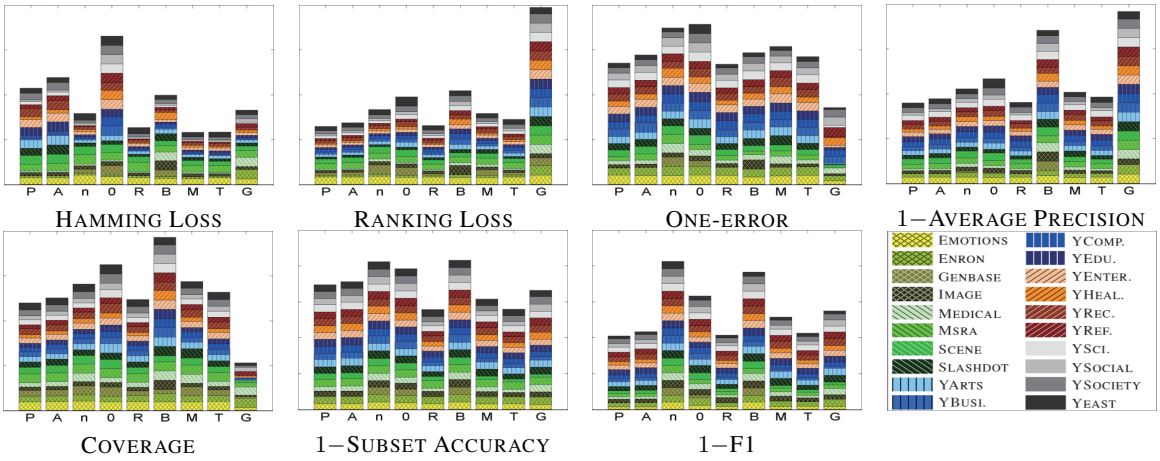
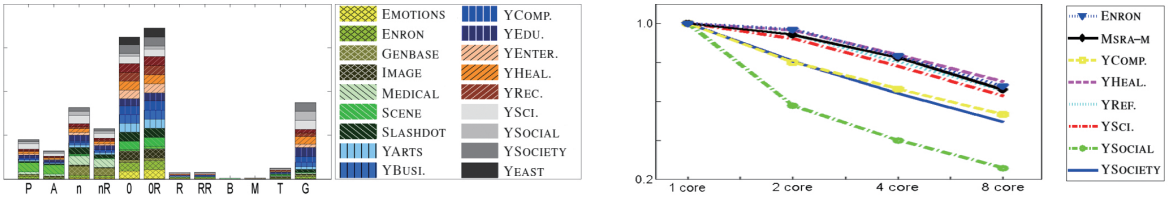


Figure 2: Comparison on the original multi-label data sets without label ordering information. Each column corresponds to an algorithm (from left to right: P: ProSVM', A: ProSVM-A', n: PCn, 0: PC0, R: RankSVM, B: BSVM, M: MLkNN, T: BoosTexter, G: GMLC). The lower the column, the better the performance.



(a) (b)

Figure 3: (a) Comparison on the *robustness* of time complexity. Each column corresponds to an algorithm (from left to right: 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). (b) Comparison on the time cost of ProSVM with multiple cores. X-axis is the number of cores. Y-axis is the time spent divided by that using only 1 core.

References

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

Bertsekas, D., and Tsitsiklis, J. 1989. *Parallel and Distributed Computation: Numerical Methods*. Prentice-Hall.

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

Boyd, S.; Parikh, N.; Chu, E.; Peleato, B.; and Eckstein, J. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning* 3(1):1–122.

Chang, C., and Lin, C. 2011. Libsvm: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* 2(3):27.

Cheng, W.; Dembczyński, K.; and Hüllermeier, E. 2010. Graded multilabel classification: The ordinal case. In *Proceedings of the 27th International Conference on Machine Learning*, 223–230.

Dekel, O.; Manning, C.; and Singer, Y. 2003. Log-linear models for label ranking. In *Advances in Neural Information Processing Systems 16*.

Dembczynski, K.; Waegeman, W.; Cheng, W.; and Hüllermeier, E. 2010. Regret analysis for performance metrics in multi-label classification: The case of hamming and subset zero-one loss. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases*, 280–295.

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

Fan, R.; Chang, K.; Hsieh, C.; Wang, X.; and Lin, C. 2008. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research* 9:1871–1874.

Forero, P.; Cano, A.; and Giannakis, G. 2010. Consensus-based distributed support vector machines. *Journal of Machine Learning Research* 11:1663–1707.

Fürnkranz, J.; Hüllermeier, E.; Mencía, E.; and Brinker, K. 2008. Multilabel classification via calibrated label ranking. *Machine Learning* 73(2):133–153.

Gärtner, T., and Vembu, S. 2010. Label ranking algorithms: A survey. In Johannes Fürnkranz, E. H., ed., *Preference Learning*. 45–64.

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

He, B., and Yuan, X. 2012. On the o(1/n) convergence rate of the douglas-rachford alternating direction method. *SIAM Journal of Numerical Analysis* 50(2):700–709.

Hüllermeier, E.; Fürnkranz, J.; Cheng, W.; and Brinker, K. 2008. Label ranking by learning pairwise preferences. *Artifical Intellengce* 172(16-17):1897–1916.

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

Kotlowski, W.; Dembczynski, K.; and Huellermeier, E. 2011. Bipartite ranking through minimization of univariate loss. In *Proceedings of the 28th International Conference on Machine Learning*, 1113–1120.

Li, H.; Wang, M.; and Hua, X.-S. 2009. Msra-mm 2.0: A large-scale web multimedia dataset. In *ICDM Workshops*, 164–169.

Qi, G.-J.; Hua, X.-S.; Rui, Y.; Tang, J.; Mei, T.; and Zhang, H.-J. 2007. Correlative multi-label video annotation. In *Proceedings of the 15th International Conference on Multimedia*, 17–26.

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

Shalev-Shwartz, S., and Singer, Y. 2006. Efficient learning of label ranking by soft projections onto polyhedra. *Journal of Machine Learning Research* 7:1567–1599.

Tsoumakas, G.; Katakis, I.; and Vlahavas, I. 2010. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*. 667–685.

Vapnik, V. N. 1998. *Statistical Learning Theory*. Wiley.

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

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

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

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

Zhang, M.-L., and Zhou, Z.-H. 2007b. Multi-label learning by instance differentiation. In *Proceedings of the 22nd AAAI Conference on Artificial Intelligence*, 669–674.

Zhang, M.; Peña, J.; and Robles, V. 2009. Feature selection for multi-label naive bayes classification. *Information Science* 179(19):3218–3229.

Zhou, Z.-H., and Yu, Y. 2005. Ensembling local learners through multimodal perturbation. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 35(4):725–735.