Privileged Multi-label Learning*

Shan You[1,2], **Chang Xu**[3], **Yunhe Wang**[1,2], **Chao Xu**[1,2], **Dacheng Tao**[3]
[1]Key Lab. of Machine Perception (MOE), School of EECS, Peking University, P. R. China
[2]Cooperative Medianet Innovation Center, Peking University, P. R. China
[3]UBTech Sydney AI Institute, School of IT, FEIT, The University of Sydney, Australia
{youshan,wangyunhe}@pku.edu.cn; {c.xu,dacheng.tao}@sydney.edu.au; chaoxu@cis.pku.edu.cn

Abstract

This paper presents privileged multi-label learning (PrML) to explore and exploit the relationship between labels in multi-label learning problems. We suggest that for each individual label, it cannot only be implicitly connected with other labels via the low-rank constraint over label predictors, but also its performance on examples can receive the explicit comments from other labels together acting as an *Oracle teacher*. We generate privileged label feature for each example and its individual label, and then integrate it into the framework of low-rank based multi-label learning. The proposed algorithm can therefore comprehensively explore and exploit label relationships by inheriting all the merits of privileged information and low-rank constraints. We show that PrML can be efficiently solved by dual coordinate descent algorithm using iterative optimization strategy with cheap updates. Experiments on benchmark datasets show that through privileged label features, the performance can be significantly improved and PrML is superior to several competing methods in most cases.

1 Introduction

Different from single-label classification, multi-label learning (MLL) allows each example to own multiple and non-exclusive labels. For instance, when to post a photo taken in the scene of Rio Olympics on Instagram, Twitter or Facebook, we may simultaneously include hashtags as #RioOlympics, #athletes, #medals and #flags. Or a related news article can be simultaneously annotated as "Sports", "Politics" and "Brazil". Multi-label learning aims to accurately allocate a group of labels to unseen examples with the knowledge harvested from the training data, and it has been widely-used in many applications, such as document categorization [Li *et al.*, 2015] and image/videos classification/annotation [Yang *et al.*, 2016].

The most straightforward approach is 1-vs-all or Binary Relevance (BR) [Tsoumakas *et al.*, 2010], which decomposes the multi-label learning into a set of independent binary classification tasks. However, due to neglecting label relationships, only passable performance can be achieved. A number of methods have thus been developed for further improving the performance by taking label relationships into consideration, such as chains of binary classification [Read *et al.*, 2011], ensemble of multi-class classification [Tsoumakas *et al.*, 2011], shared parameters [Liu *et al.*, 2017] and label-specific features [Zhang and Wu, 2015; Xu *et al.*, 2015]. Recently, embedding-based methods have emerged as a mainstream solution of the multi-label learning problem. The approaches assume that the label matrix is low-rank, and adopt different manipulations to embed the original label vectors, such as principal component analysis [Tai and Lin, 2012], latent representation [Xu *et al.*, 2016a], and manifold deduction [Bhatia *et al.*, 2015; Hou *et al.*, 2016].

Most of low-rank based multi-label learning algorithms exploit label relationships in the hypothesis space. The hypotheses of different labels are interacted with each other under the low-rank constraint, which is as an implicit use of label relationships. By contrast, multiple labels can help each other in a more explicit way, where the hypothesis of a label is not only evaluated by the label itself, but also can be assessed by the other labels. More specifically in multi-label learning, for the label hypothesis at hand, the other labels can together act as an *Oracle teacher* to provide some *comments* on its performance, which is then beneficial for updating the learner. Multiple labels of examples can only be accessed in the training stage instead of the testing stage, and then Oracle teachers only exist in the training stage. This *privileged* setting has been studied in LUPI (learning using privileged information) paradigm [Vapnik *et al.*, 2009; Vapnik and Vashist, 2009; Vapnik and Izmailov, 2015] and it has been reported that appropriate privileged information can boost the performance in ranking [Sharmanska *et al.*, 2013] and classification [Pechyony and Vapnik, 2010].

In this paper, we bridge connections between labels through privileged label information and then formulate an effective privileged multi-label learning (PrML) method. For each label, each example's privileged label feature can be generated from other labels. Then it is able to provide additional guidance on the learning of this label, given the underlying connections between labels. By integrating the privileged information into

*This work was supported by NSFC 61375026 and 2015BAF15B00, and ARC Projects: ARC FT-130101457, DP-140102164, LP-150100671.

the low-rank based multi-label learning, each label predictor learned from the resulting model not only interacts with other labels via their predictors, but also receives explicit comments from these labels. Iterative optimization strategy is employed to solve PrML, and we theoretically show that each subproblem can be solved by dual coordinate descent algorithm with the guarantee of solution's uniqueness. Experimental results demonstrate the significance of exploiting the privileged label features and the effectiveness of the proposed algorithm.

2 Problem Formulation

In this section we elaborate the intrinsic privileged information in multi-label learning and formulate the corresponding privileged multi-label learning (PrML) as well.

We first introduce multi-label learning (MLL) problem and its frequent notations. Given n training points, we denote the whole data set as $\mathcal{D} = \{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_n, \mathbf{y}_n)\}$, where $\mathbf{x}_i \in \mathcal{X} \subseteq \mathbb{R}^d$ is the input feature vector and $\mathbf{y}_i \in \mathcal{Y} \subseteq \{-1, 1\}^L$ is the corresponding label vector with the label size L. Let $X = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in \mathbb{R}^{d \times n}$ be the data matrix and $Y = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n] \in \{-1, 1\}^{L \times n}$ be the label matrix. Specifically, $Y_{ij} = 1$ if and only if the i-th label is assigned to the example $\mathbf{x}_j$ and $Y_{ij} = -1$ otherwise. Given the dataset $\mathcal{D}$, multi-label learning is formulated as learning a mapping function $f : \mathbb{R}^d \to \{-1, 1\}^L$ that can accurately predict labels for unseen test points.

2.1 Low-rank Multi-label Embedding

A straightforward manner to parameterize the decision function is using linear classifiers, *i.e.* $f(\mathbf{x}) = Z^T\mathbf{x} = [\mathbf{z}_1, ..., \mathbf{z}_L]^T\mathbf{x}$ where $Z \in \mathbb{R}^{d \times L}$. Note that the linear form is actually incorporated with the bias term by augmenting an additional 1 to the feature vector $\mathbf{x}$. Binary Relevance (BR) method [Tsoumakas *et al.*, 2010] decomposes multi-label learning into a set of single-label learning problems. The binary classifier for each label can be obtained by the widely-used SVM method:

$$\begin{aligned}\min_{\mathbf{z}_i=[\mathbf{z}_i^*;b_i],\boldsymbol{\xi}_i} \quad & \frac{1}{2}\|\mathbf{z}_i^*\|_2^2 + C\sum_{j=1}^{n}\xi_{ij} \\ \text{s.t.} \quad & Y_{ij}(\langle \mathbf{z}_i^*, \mathbf{x}_j\rangle + b_i) \geq 1 - \xi_{ij} \\ & \xi_{ij} \geq 0, \forall j = 1, ..., n,\end{aligned} \tag{1}$$

where $\boldsymbol{\xi}_i = [\xi_{i1}, ..., \xi_{in}]^T$ is slack variable and $\langle\cdot\rangle$ is the inner product between two vectors or matrices. Predictors $\{\mathbf{z}_1, ..., \mathbf{z}_L\}$ of different labels are thus independently solved without considering relationships between labels, which limits the classification performance of BR method.

Some labels can be closely connected and used to occur together on examples, and thus the label matrix is often supposed to be low-rank, which leads to the low rank of label predictor matrix $Z = [\mathbf{z}_1, ..., \mathbf{z}_L]$ as a result. Considering the rank of Z as k, which is smaller than d and L, we are able to employ two smaller matrices to approximate Z, *i.e.* $Z = D^TW$. $D \in \mathbb{R}^{k \times d}$ can be seen as a dictionary of hypotheses in latent space $\mathbb{R}^k$, while each $\boldsymbol{w}_i$ in $W = [\boldsymbol{w}_1, ..., \boldsymbol{w}_L] \in \mathbb{R}^{k \times L}$ is the coefficient vector to generate the predictor of i-th label from the hypothesis dictionary D. Each classifier $\mathbf{z}_i$ is represented as $\mathbf{z}_i = D^T\boldsymbol{w}_i$ $(i = 1, 2, ..., L)$ and Problem (1) can be extended into:

$$\begin{aligned}\min_{D,W,\xi} \quad & \frac{1}{2}(\|D\|_F^2 + \sum_{i=1}^{L}\|\boldsymbol{w}_i\|_2^2) + C\sum_{i=1}^{L}\sum_{j=1}^{n}\xi_{ij} \\ \text{s.t.} \quad & Y_{ij}(\langle D^T\boldsymbol{w}_i, \mathbf{x}_j\rangle) \geq 1 - \xi_{ij} \\ & \xi_{ij} \geq 0, \forall i = 1, ..., L; j = 1, ..., n,\end{aligned} \tag{2}$$

where $\xi = [\boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_L]^T$. Thus in Eq.(2), the classifiers of all labels $\mathbf{z}_i$ are drawn from an identical low-dimensional subspace, *i.e.* the row space of D. Then using block coordinate descent, either D or W can be solved within the empirical risk minimization (ERM) framework by turning it into a hinge loss minimization problem.

2.2 Privileged Information in Multi-label Learning

The slack variable ξ_{ij} in Eq.(2) indicates the prediction error of the j-th example on the i-th label. In fact, it depicts the error-tolerant ability of a model, and is directly related to the optimal classifier and its classification performance. From a different point of view, slack variables can be regarded as *comments* of some *Oracle Teacher* on the performance of predictors on each example. In multi-label context for each label, its hypothesis is not only evaluated by itself, but also assessed by the other labels. Thus other labels can be seen as its Oracle teacher, who will provide some comments during this label's learning. Note that these label values are known as a priori only during training; when we get down to learning the i-th label's predictor, we actually know the values of other labels for each training point $\mathbf{x}_j$. Therefore, we can formulate the other label values as privileged information (or hidden information) of each example. Let

$$\tilde{\mathbf{y}}_{i,j} \triangleq \mathbf{y}_j, \text{ with } i\text{-th element being } 0. \tag{3}$$

We call $\tilde{\mathbf{y}}_{i,j}$ the training point $\mathbf{x}_j$'s *privileged label feature* on the i-th label. It can be seen that the privileged label space is constructed straightforwardly from the original label space. These privileged label features can thus be regarded as an explicit way to connect all labels. In addition, note that the *valid* dimension (removing 0) of $\tilde{\mathbf{y}}_{i,j}$ is $L - 1$, since we take the other $L - 1$ label values as the privileged label features. Moreover, not all the other labels have the positive impact on the learning of some label [Sun *et al.*, 2014], and thus it is appropriate to strategically select some key labels to formulate the privileged label features. We will discuss this in the Experiment section.

Since for each label, the other labels serve as the Oracle teacher via the privileged label feature $\tilde{\mathbf{y}}_{i,j}$ on each example, the comments on slack variables can be modelled as a linear function [Vapnik and Vashist, 2009],

$$\xi_{ij}(\tilde{\mathbf{y}}_{i,j}; \tilde{\boldsymbol{w}}_i) = \langle \tilde{\boldsymbol{w}}_i, \tilde{\mathbf{y}}_{i,j}\rangle. \tag{4}$$

The function $\xi_{ij}(\tilde{\mathbf{y}}_{i,j}; \tilde{\boldsymbol{w}}_i)$ is thus called *correcting function* with respect to the i-th label, where $\tilde{\boldsymbol{w}}_i$ is the parameter vector. As shown in Eq.(4), the privileged comments $\tilde{\mathbf{y}}_{i,j}$ directly correct the values of slack variables as the prior knowledge or the additional information. Integrating privileged features as Eq.(4) into the SVM stimulates the popular SVM+ method [Vapnik and Vashist, 2009], which has been proved to improve the convergence rate and the performance.

Integrating the proposed privileged label features into the low-rank parameter structure as Eqs.(2) and (4), we formulate a new multi-label learning model, privileged multi-label learning (PrML) by casting it into the SVM+-based LUPI paradigm,

$$\begin{aligned} \min_{D,W,\tilde{W}} \quad & \frac{1}{2}\|D\|_F^2 + \frac{1}{2}\sum_{i=1}^{L}(\gamma_1\|\boldsymbol{w}_i\|_2^2 + \gamma_2\|\tilde{\boldsymbol{w}}_i\|_2^2) \\ & + C\sum_{i=1}^{L}\sum_{j=1}^{n}\langle\tilde{\boldsymbol{w}}_i, \tilde{\mathbf{y}}_{i,j}\rangle \\ \text{s.t.} \quad & Y_{ij}\langle D^T\boldsymbol{w}_i, \mathbf{x}_j\rangle \geq 1 - \langle\tilde{\boldsymbol{w}}_i, \tilde{\mathbf{y}}_{i,j}\rangle \\ & \langle\tilde{\boldsymbol{w}}_i, \tilde{\mathbf{y}}_{i,j}\rangle \geq 0, \forall i = 1, ..., L; j = 1, ..., n, \end{aligned} \tag{5}$$

where $\tilde{W} = [\tilde{\boldsymbol{w}}_1, ..., \tilde{\boldsymbol{w}}_L]$. Particularly, we absorb the bias term to obtain a compact variant of the original SVM+, because it is turned out to have a simpler form in the dual space and can be solved more efficiently. In this way, the training data within multi-label learning is actually in the triplet fashion, *i.e.* $(\mathbf{x}_i, \mathbf{y}_i, \tilde{Y}_i), i = 1, ..., n$, where $\tilde{Y}_i = [\tilde{\mathbf{y}}_{1,i}, ..., \tilde{\mathbf{y}}_{L,i}]$ is the privileged label feature matrix for each label.

Remark. When $W = I$, *i.e.* the low-dimensional projection is identical, the proposed PrML degenerates into a simpler BR-style model (we call it privileged Binary Relevance, PrBR), where the whole model decomposes into L independent binary models. However, every single model is still combined with the comments form privileged information, thus it may still be superior to BR.

3 Optimization

In this section, we present how to solve the proposed algorithm Eq.(5). The whole model of Eq.(5) is not convex due to the multiplication of $D^T\boldsymbol{w}_i$ in constraints. However, each subproblem with fixed D or W is convex. Note that $\langle D^T\boldsymbol{w}_i, \mathbf{x}_j\rangle$ has two equivalent forms, *i.e.* $\langle\boldsymbol{w}_i, D\mathbf{x}_j\rangle$ and $\langle D, \boldsymbol{w}_i\mathbf{x}_j^T\rangle$, and thus the correcting function can be coupled with D or $\tilde{W}$, without damaging the convexity of either subproblem. In this way, Eq.(5) can be solved using the alternative iteration strategy, *i.e.* iteratively conducting the following two steps: optimizing W and privileged variable $\tilde{W}$ with fixed D, and updating D and privileged variable $\tilde{W}$ with fixed W. Both subproblems are related to SVM+, inducing their dual problems to be quadratic programming (QP). In the following, we elaborate the solving process in real implementations.

3.1 Optimizing $W, \tilde{W}$ with Fixed D

Fixing D, Eq.(5) can be decomposed into L independent binary classification problems, each of which regards the variable pair $(\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i)$. Parallel techniques or multi-core computation can thus be employed to speed up the training process. In specific, the optimization problem with respect to $(\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i)$ is

$$\begin{aligned} \min_{\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i} \; & \frac{1}{2}(\gamma_1\|\boldsymbol{w}_i\|_2^2 + \gamma_2\|\tilde{\boldsymbol{w}}_i\|_2^2) + C\sum_{j=1}^{n}\langle\tilde{\boldsymbol{w}}_i, \tilde{\mathbf{y}}_{i,j}\rangle \\ \text{s.t.} \; & Y_{ij}\langle\boldsymbol{w}_i, D\mathbf{x}_j\rangle \geq 1 - \langle\tilde{\boldsymbol{w}}_i, \tilde{\mathbf{y}}_{i,j}\rangle \\ & \langle\tilde{\boldsymbol{w}}_i, \tilde{\mathbf{y}}_{i,j}\rangle \geq 0, \forall j = 1, ..., n. \end{aligned} \tag{6}$$

and its dual form is

$$\begin{aligned} \max_{\boldsymbol{\alpha}, \boldsymbol{\beta}} \; & -\frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y}_i^*)^T K_D (\boldsymbol{\alpha} \circ \mathbf{y}_i^*) + \mathbf{1}^T\boldsymbol{\alpha} \\ & - \frac{1}{2\gamma}(\boldsymbol{\alpha} + \boldsymbol{\beta} - C\mathbf{1})^T \tilde{K}_i (\boldsymbol{\alpha} + \boldsymbol{\beta} - C\mathbf{1}) \end{aligned} \tag{7}$$

with the parameter update $\gamma \leftarrow \gamma_2/\gamma_1, C \leftarrow C/\gamma_1$ and the constraints $\boldsymbol{\alpha} \succeq 0, \boldsymbol{\beta} \succeq 0$, *i.e.* $\alpha_j \geq 0, \beta_j \geq 0, \forall j \in [1 : n]$. Moreover, $\mathbf{y}_i^* = [Y_{i1}, Y_{i2}, ..., Y_{in}]^T$ is the label-wise vectors for the i-th label. $\circ$ is the Hadamard (element-wise) product of two vectors or matrices. $K_D \in \mathbb{R}^{n \times n}$ is the D-based features' inner product (kernel) matrix with $K_D(j, q) = \langle D\mathbf{x}_j, D\mathbf{x}_q\rangle$. $\tilde{K}_i$ is the privileged label features' inner product (kernel) matrix with respect to the i-th label, where $\tilde{K}_i(j, q) = \langle\tilde{\mathbf{y}}_{i,j}, \tilde{\mathbf{y}}_{i,q}\rangle$. $\mathbf{1}$ is the vector with all ones.

[Pechyony *et al.*, 2010] proposed an SMO-style algorithm (gSMO) for SVM+ problem. However, because of the bias term, the Lagrange multipliers are tangled together in the dual problem, which leads to a more complicated constraint set

$$\{(\boldsymbol{\alpha}, \boldsymbol{\beta}) | \boldsymbol{\alpha}^T\mathbf{y}_i^* = 0, \mathbf{1}^T(\boldsymbol{\alpha} + \boldsymbol{\beta} - C\mathbf{1}) = 0, \boldsymbol{\alpha} \succeq 0, \boldsymbol{\beta} \succeq 0\}$$

than $\{(\boldsymbol{\alpha}, \boldsymbol{\beta}) | \boldsymbol{\alpha} \succeq 0, \boldsymbol{\beta} \succeq 0\}$ in our PrML. Hence by absorbing the bias term, Eq.(6) can produce a more compact dual problem only with non-negative constraint. Thus this dual QP problem [Zhang *et al.*, 2017] can use coordinate descent (CD)[1] algorithm, and a closed-form solution can be obtained in each iteration step [Li *et al.*, 2016]. After solving the Eq.(7), according to the Karush-Kuhn-Tucker (KKT) conditions, the optimal solution for the primal problem (6) can be expressed by the Lagrange multipliers:

$$\begin{aligned} \boldsymbol{w}_i &= \textstyle\sum_{j=1}^{n}\alpha_j Y_{ij} D\mathbf{x}_j \\ \tilde{\boldsymbol{w}}_i &= \frac{1}{\gamma}\textstyle\sum_{j=1}^{n}(\alpha_j + \beta_j - C)\tilde{\mathbf{y}}_{i,j} \end{aligned} \tag{8}$$

3.2 Optimizing $D, \tilde{W}$ with Fixed W

Given fixed coefficient matrix W, we update and learn the linear transformation D with the help of comments provided by privileged information. Thus the problem (5) for $(D, \tilde{W})$ is reduced to

$$\begin{aligned} \min_{D, \tilde{W}} \; & \frac{1}{2}\|D\|_F^2 + \frac{\gamma_2}{2}\sum_{i=1}^{L}\|\tilde{\boldsymbol{w}}_i\|_2^2 + C\sum_{i=1}^{L}\sum_{j=1}^{n}\langle\tilde{\boldsymbol{w}}_i, \tilde{\mathbf{y}}_{i,j}\rangle \\ \text{s.t.} \; & Y_{ij}\langle D, \boldsymbol{w}_i\mathbf{x}_j^T\rangle \geq 1 - \langle\tilde{\boldsymbol{w}}_i, \tilde{\mathbf{y}}_{i,j}\rangle \\ & \langle\tilde{\boldsymbol{w}}_i, \tilde{\mathbf{y}}_{i,j}\rangle \geq 0, \forall i = 1, ..., L; j = 1, ..., n. \end{aligned} \tag{9}$$

Eq.(9) has Ln constraints, each of which can be indexed with a two-dimensional subscript $[i, j]$. The Lagrange multipliers

[1] We optimize an equivalent "min" problem instead of the original "max" one.

Algorithm 1 Privileged Multi-label Learning (PrML)

```
Input: Training data: feature matrix $X = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n] \in \mathbb{R}^{d \times n}$, label matrix $Y = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_n] \in \{-1, 1\}^{L \times n}$.
    Learning parameters: $\gamma_1, \gamma_2, C \geq 0$.
 1: Construction of privileged label features for each label and each training point, e.g. as Eq.(3).
 2: initialization of $D$
 3: while not convergence do
 4:     for each $i \in [1 : L]$ do
 5:         $[\boldsymbol{\alpha}, \boldsymbol{\beta}] \leftarrow$ solving Eq.(7)
 6:         update $\boldsymbol{w}_i, \tilde{\boldsymbol{w}}_i$ according to Eq.(8)
 7:     end for
 8:     $[\boldsymbol{\alpha}, \boldsymbol{\beta}] \leftarrow$ solving Eq.(10)
 9:     update $D, \tilde{W}$ according to Eq.(11)
10: end while
Output: A linear multi-label classifier $Z = D^T W$, together with a correcting function $\tilde{W}$ w.r.t. $L$ labels.
```

of Eq.(9) are thus two-dimensional as well. To make the dual problem of Eq.(9) consistent with Eq.(7), we define a bijection $\phi : [1 : L] \times [1 : n] \to [1 : Ln]$ as the row-based vectorization index mapping, *i.e.* $\phi([i, j]) = (i - 1)n + j$. In a nutshell, we arrange the constraints (also the multipliers) according to the order of row-based vectorization. In this way, the corresponding dual problem of Eq.(9) is formulated as

$$\begin{aligned} \max_{\boldsymbol{\alpha} \succeq 0, \boldsymbol{\beta} \succeq 0} \quad & -\frac{1}{2}(\boldsymbol{\alpha} \circ \mathbf{y}^*)^T K_W (\boldsymbol{\alpha} \circ \mathbf{y}^*) + \mathbf{1}^T \boldsymbol{\alpha} \\ & -\frac{1}{2\gamma_2}(\boldsymbol{\alpha} + \boldsymbol{\beta} - C\mathbf{1})^T \tilde{K} (\boldsymbol{\alpha} + \boldsymbol{\beta} - C\mathbf{1}) \end{aligned} \tag{10}$$

where $\mathbf{y}^* = [\mathbf{y}_1^*; \mathbf{y}_2^*; ...; \mathbf{y}_L^*]$ is the row-based vectorization of Y and $\tilde{K} = diag(\tilde{K}_1, \tilde{K}_2, ..., \tilde{K}_L)$ is a block diagonal matrix, which corresponds to the kernel matrix of privileged label features. K_W is the kernel matrix of input features with every element $K_W(s, t) = \langle G_{\phi^{-1}(s)}, G_{\phi^{-1}(t)} \rangle$, where $G_{ij} = \boldsymbol{w}_i \mathbf{x}_j^T$. Based on the KKT conditions, $(D, \tilde{W})$ can be constructed using $(\boldsymbol{\alpha}, \boldsymbol{\beta})$:

$$\begin{aligned} D &= \textstyle\sum_{s=1}^{Ln} \alpha_s y_s^* G_{\phi^{-1}(s)} \\ \tilde{\boldsymbol{w}}_i &= \textstyle\frac{1}{\gamma_2} \sum_{j=1}^{n} (\alpha_{\phi([i,j])} + \beta_{\phi([i,j])} - C) \tilde{\mathbf{y}}_{i,j} \end{aligned} \tag{11}$$

In this way, Eq.(10) has an identical optimization form with Eq.(7). Thus we can also turn it to the fast CD method [Li *et al.*, 2016]. However, due to the script index mapping, directly using the method proposed in [Li *et al.*, 2016] is very expensive. Some modification is required for adaption and acceleration, such as applying the block sparsity of the privileged kernel matrix $\tilde{K}$. Also note that one primary merit of this algorithm is the free calculation of the whole kernel matrix. Instead, we only need to calculate its diagonal elements.

3.3 Framework of PrML

Our proposed privileged multi-label learning is summarized in Algorithm 1. As indicated in Algorithm 1, both D and W are updated with the help of comments from privileged information. Note that the primal variables and dual variables are connected with KKT connections, and thus in real applications lines 5-6 and 8-9 in Algorithm 1 can be implemented iteratively. Since each subproblem is actually a linear SVM+ optimization and solved by the CD method, its convergence is consistent with that of the dual CD algorithm for linear SVM [Hsieh *et al.*, 2008]. Due to the cheap updates, [Hsieh *et al.*, 2008; Li *et al.*, 2016] empirically showed it can be much faster than GMO-style methods and many other convex solvers when d (number of features) is large. Moreover, the independence of labels in Problem (6) enables to use parallel techniques and multicore computation to accommodate the large L (number of labels). As for a large n (number of examples) (also large L for Problem (9)), we can use mini-batch CD method [Takac *et al.*, 2015] , where each time a batch of examples are selected and CD updates are parallelly applied to them, *i.e.* lines 5-17 can be implemented parallelly. Also recently [Chiang *et al.*, 2016] designed a framework for parallel CD and achieved significant speeding up even when the d and n are very large. Thus, our model can scale to d, L and n. In addition, the solution for each of subproblem is also unique, as Theorem 1 stated.

Table 1: Data statistics. n is the total number of examples. d and L are the number of features and labels, respectively; $\bar{L}$ and Den(L) are the average number of positive labels in an instance and the label density, respectively. 'Type' means feature type.

Dataset	n	d	L	$\bar{L}$	Den(L)	type
enron	1702	1001	53	3.378	0.064	nominal
yeast	2417	103	14	4.237	0.303	numeric
corel5k	5000	499	374	3.522	0.009	nominal
bibtex	7395	1836	159	2.402	0.015	nominal
eurlex	19348	5000	3993	5.310	0.001	nominal
mediamill	43907	120	101	4.376	0.043	numeric

Theorem 1. *The solution to the problem (6) or (9) is unique for any* $\gamma_1 > 0, \gamma_2 > 0, C > 0$.

Proof of Theorem 1 mainly lies in the strict convexity of the objective function in either Eq.(6) or (9). In this way, the correcting function $\tilde{W}$ serves as a bridge to channel the D and W, and the convergence of $\tilde{W}$ infers the convergence of D and W. Thus we can take $\tilde{W}$ as the barometer of the whole algorithm's convergence.

4 Experimental Results

In this section, we conduct various experiments on benchmark datasets to validate the effectiveness of using the intrinsic privileged information for multi-label learning. In addition, we also investigate the performance and superiority of the proposed PrML model comparing to recent competing multi-label methods.

4.1 Experiment Configuration

Datasets. We select six benchmark multi-label datasets, including enron, yeast, corel5k, bibtex, eurlex and mediamill. Specially, we consider the cases when d (eurlex), L (eurlex) and n (corel5k, bibtex, eurlex & mediamill) are large respectively. Also note that enron, corel5k, bibtex and eurlex are of sparse features. See Table 1 for the details of these datasets.

Comparison approaches.

1). BR (binary relevance) [Tsoumakas *et al.*, 2010]. An SVM

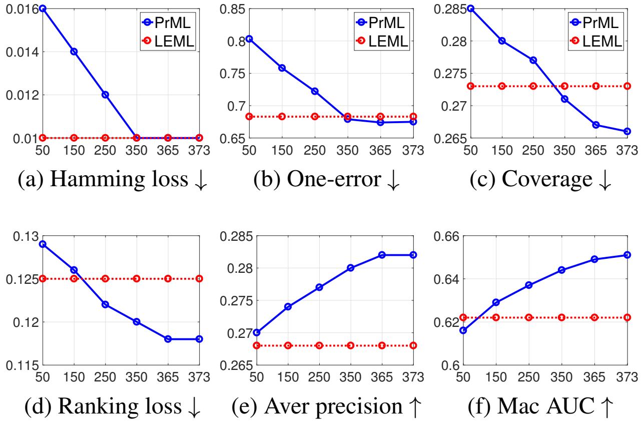
is trained with respect to each label.
2). ECC (ensembles of classifier chains) [Read *et al.*, 2011]. It turns ML into a series of binary classification problems.
3). RAKEL (random k-labelsets) [Tsoumakas *et al.*, 2011]. It transforms MLL into an ensemble of multi-class classification problems.
4). LEML (low rank empirical risk minimization for multi-label learning) [Yu *et al.*, 2014]. It is a low-rank embedding approach which is casted into ERM framework.
5). ML^2 (multi-label manifold learning) [Hou *et al.*, 2016]. It is a latest multi-label learning method, which is based on the manifold assumption in label space.

Evaluation Metrics. We use six prevalent metrics to evaluate the performance of all methods, including Hamming loss, One-error, Coverage, Ranking loss, Average precision (Aver precision) and Macro-averaging AUC (Mac AUC). Note that all evaluation metrics have the value range [0,1]. In addition, for the first four metrics, the smaller values would indicate the better classification performance and we use ↓ to index this positive logic. On the contrary, for the last two metrics larger values represent the better performance, indexed by ↑.

4.2 Validation of Privileged Label Features

We first validate the effectiveness of the proposed privileged information for multi-label learning. As discussed previously, the privileged label features serve as an guidance or comments from an Oracle teacher to connect the learning of all the labels. For the sake of fairness, we simply implement the validation experiments with LEML (without privileged label features) and PrML (with privileged label features). Note that our proposed privileged label features are composed with the values of labels; however, not all labels have prominent connections in multi-label learning [Sun *et al.*, 2014]. Thus we selectively construct the privileged label features with respect to each label.

Particularly, we just use K-nearest neighbor rule to form the pool per label. For each label, only labels in its label pool, instead of the whole label set, are reckoned to provide mutual guidance during its learning. In our implementation, we simply utilize Hamming distance to accomplish search of K-nearest neighbor on the dataset corel5k. Particularly, we randomly selected 50% examples without repeating as the training set and the rest ones as the testing set. In our experiment, parameter γ_1 is set to be 1; γ_2 and C are in the range of $0.1 \sim 2.0$ and $10^{-3} \sim 20$ respectively, and determined using cross validation by a part of training points. Both algorithms have the same embedding dimension $k = \lceil 0.9L \rceil$, where $\lceil r \rceil$ is the smallest integer greater than r. Moreover, we carry out independent tests ten times and the average results are shown in Figure 1.



(a) Hamming loss ↓ (b) One-error ↓ (c) Coverage ↓

(d) Ranking loss ↓ (e) Aver precision ↑ (f) Mac AUC ↑

Figure 1: Classification results of PrML (blue solid line) and LEML (red dashed line) on corel5k (50% for training & 50% for testing) *w.r.t.* different dimension of privileged label features. In each subfigure, the horizontal axis represents the number of privileged label features while the vertical axis indicates the corresponding metric values.

As shown in Figure 1, we have the following two observations. (a) PrML is clearly superior to LEML when we select enough labels as privileged label features, *e.g.* more than 350 labels in corel5k dataset. Since their only difference lies in the usage of the privileged information, we can conclude that the guidance from the privileged information, *i.e.* the proposed privileged label features, can significantly improve the performance of multi-label learning. (b) With more labels involved in the privileged label features, the performance of PrML keeps improving in a steady speed, and when the dimension of privileged label features is large enough, the performance tends to stabilize on the whole.

The number of labels is directly related to the complexity of correcting function defined as a linear function. Thus few labels might induce the low function complexity, and the correcting function can not determine the optimal slack variables. In this way, the fault-tolerant capacity would be crippled and thus the performance is even worse than LEML. For example, when the dimension of privileged labels is less than 250 on corel5k, the Hamming loss, One-error, Coverage and Ranking loss of PrML is much larger than LEML. In contrast, overmuch labels might introduce unnecessary guidance of labels, and the *extra* labels thus make no contribution to the further improvement of classification performance. For instance, the performance with 365 labels involved in privileged label features would be on par with that of all the other (373) labels in Hamming loss, One-error, Ranking loss and Average precision. Moreover, in real applications, it is still a safe choice that all other labels are involved in privileged information.

4.3 Performance Comparison

Now we formally analyze the performance of our proposed privileged multi-label learning (PrML) in comparison with popular state-of-the-art methods. For each dataset, we randomly selected 50% examples without repeating as the training set and the rest for testing. For the results' credibility, the dataset division process is implemented ten times independently and we recorded the corresponding results in each trail. Parameters γ_1, γ_2 and C are determined in the same manner as before. As for the low embedding dimension k, following the wisdom of [Yu *et al.*, 2014], we choose k to be in $\{\lceil 0.8L \rceil, \lceil 0.85L \rceil, \lceil 0.9L \rceil, \lceil 0.95L \rceil\}$ and determined by cross validation using a part of training points. Particularly, we also cover the PrBR (privileged information + BR) to further investigate the proposed privileged information. The detailed results are reported in Table 2.

Table 2: Average predictive performance (mean ± std. deviation) of ten indepedent trails for various multi-label learning methods. In each trail, 50% examples are randomly selected without repeating as training set and the rest as testing set. The top performance among all methods is marked in boldface.

dataset	method	Hamming loss ↓	One-error ↓	Coverage ↓	Ranking loss ↓	Aver precision ↑	Mac AUC ↑
enron	BR	0.060±0.001	0.498±0.012	0.595±0.010	0.308±0.007	0.449±0.011	0.579±0.007
	ECC	0.056±0.001	0.293±0.008	0.349±0.014	0.133±0.004	0.651±0.006	0.646±0.008
	RAKEL	0.058±0.001	0.412±0.016	0.523±0.008	0.241±0.005	0.539±0.006	0.596±0.007
	LEML	**0.049±0.002**	0.320±0.004	0.276±0.005	0.117±0.006	0.661±0.004	0.625±0.007
	ML^2	0.051±0.001	**0.258±0.090**	0.256±0.017	0.090±0.012	0.681±0.053	**0.714±0.021**
	PrBR	0.053±0.001	0.342±0.010	0.238±0.006	**0.088±0.003**	0.618±0.004	0.638±0.005
	PrML	0.050±0.001	0.288±0.005	**0.221±0.005**	**0.088±0.006**	**0.685±0.005**	0.674±0.004
yeast	BR	0.201±0.003	0.256±0.008	0.641±0.005	0.315±0.005	0.672±0.005	0.565±0.003
	ECC	0.207±0.003	0.244±0.009	0.464±0.005	0.186±0.003	0.752±0.006	0.646±0.003
	RAKEL	0.202±0.003	0.251±0.008	0.558±0.006	0.245±0.004	0.720±0.005	0.614±0.003
	LEML	0.201±0.004	0.224±0.003	0.480±0.005	0.174±0.004	0.751±0.006	0.642±0.004
	ML^2	**0.196±0.003**	0.228±0.009	**0.454±0.004**	0.168±0.003	0.765±0.005	**0.702±0.007**
	PrBR	0.227±0.004	0.237±0.006	0.487±0.005	0.204±0.003	0.719±0.005	0.623±0.004
	PrML	0.201±0.003	**0.214±0.005**	0.459±0.004	**0.165±0.003**	**0.771±0.003**	0.685±0.003
corel5k	BR	0.012±0.001	0.849±0.008	0.898±0.003	0.655±0.004	0.101±0.003	0.518±0.001
	ECC	0.015±0.001	0.699±0.006	0.562±0.007	0.292±0.003	0.264±0.003	0.568±0.003
	RAKEL	0.012±0.001	0.819±0.010	0.886±0.004	0.627±0.004	0.122±0.004	0.521±0.001
	LEML	**0.010±0.001**	0.683±0.006	0.273±0.008	0.125±0.003	0.268±0.005	0.622±0.006
	ML^2	**0.010±0.001**	**0.647±0.007**	0.372±0.006	0.163±0.003	**0.297±0.002**	**0.667±0.007**
	PrBR	**0.010±0.001**	0.740±0.007	0.367±0.005	0.165±0.004	0.227±0.004	0.560±0.005
	PrML	**0.010±0.001**	0.675±0.003	**0.266±0.007**	**0.118±0.003**	0.282±0.005	0.651±0.004
bibtex	BR	0.015±0.001	0.559±0.004	0.461±0.006	0.303±0.004	0.363±0.004	0.624±0.002
	ECC	0.017±0.001	0.404±0.003	0.327±0.008	0.192±0.003	0.515±0.004	0.763±0.003
	RAKEL	0.015±0.001	0.506±0.005	0.443±0.006	0.286±0.003	0.399±0.004	0.641±0.002
	LEML	0.013±0.001	0.394±0.004	0.144±0.002	0.082±0.003	0.534±0.002	0.757±0.003
	ML^2	0.013±0.001	**0.365±0.004**	**0.128±0.003**	0.067±0.002	**0.596±0.004**	**0.911±0.002**
	PrBR	0.014±0.001	0.426±0.004	0.178±0.010	0.096±0.005	0.529±0.009	0.702±0.003
	PrML	**0.012±0.001**	0.367±0.003	0.131±0.007	**0.066±0.003**	0.571±0.004	0.819±0.005
eurlex	BR	0.018±0.004	0.537±0.002	0.322±0.008	0.186±0.009	0.388±0.005	0.689±0.007
	ECC	0.011±0.003	0.492±0.003	0.298±0.004	0.155±0.006	0.458±0.004	0.787±0.009
	RAKEL	0.009±0.004	0.496±0.007	0.277±0.009	0.161±0.001	0.417±0.010	0.822±0.005
	LEML	0.003±0.002	0.447±0.005	0.233±0.003	0.103±0.010	0.488±0.006	0.821±0.014
	ML^2	**0.001±0.001**	0.320±0.001	**0.171±0.003**	**0.045±0.007**	0.497±0.003	0.885±0.003
	PrBR	0.007±0.008	0.484±0.003	0.229±0.009	0.108±0.009	0.455±0.003	0.793±0.008
	PrML	**0.001±0.002**	**0.299±0.003**	0.192±0.008	0.057±0.002	**0.526±0.009**	**0.892±0.004**
mediamill	BR	0.031±0.001	0.200±0.003	0.575±0.003	0.230±0.001	0.502±0.002	0.510±0.001
	ECC	0.035±0.001	0.150±0.005	0.467±0.009	0.179±0.008	0.597±0.014	0.524±0.001
	RAKEL	0.031±0.001	0.181±0.002	0.560±0.002	0.222±0.001	0.521±0.001	0.513±0.001
	LEML	0.030±0.001	**0.126±0.003**	0.184±0.007	0.084±0.004	0.720±0.007	0.699±0.010
	ML^2	0.035±0.002	0.231±0.004	0.278±0.003	0.121±0.003	0.647±0.002	**0.847±0.003**
	PrBR	0.031±0.001	0.147±0.005	0.255±0.003	0.092±0.002	0.648±0.003	0.641±0.004
	PrML	**0.029±0.002**	0.130±0.002	**0.172±0.004**	**0.055±0.006**	**0.726±0.002**	0.727±0.008

From Table 2, we can see the proposed PrML is comparable to the state-of-the-art ML^2 method, and significantly surpasses the other competing multi-label methods. Concretely, across all evaluation metrics and datasets, PrML ranks first in 52.8% cases and the first two in all cases; even in the second place, PrML's performance is close to the top one. Comparing BR & PrBR, and LEML & PrML, we can safely infer that the privileged information plays an important role in enhancing the classification performance of multi-label predictors. Besides, in all the 36 cases, PrML wins 34 cases against PrBR and plays a tie twice in Ranking loss on enron and Hamming loss on corel5k respectively, which implies that the low-rank structure in PrML has positive impact in further improving the multi-label performance. Therefore, we can see PrML has inherited the merits of both low-rank parameter structure and privileged label information. In addition, PrML and LEML tend to perform better on datasets with more labels (>100). This might be because the low-rank assumption is more sensible when the number of labels is considerably large.

5 Conclusion

In this paper, we investigate the intrinsic privileged information to connect labels in multi-label learning. Tactfully, we regard the label values as the privileged label features. This strategy indicates that for each label's learning, other labels of each example may serve as its Oracle comments on the learning of this label. Then we propose to actively construct privileged label features directly from the label space. During the optimization, both the dictionary D and the coefficient matrix W can receive the comments from the privileged information. Experimental results show that with this very privileged information, the classification performance can be significantly improved. Thus we can also take the privileged label features as a way to boost the classification performance of the low-rank based models.

As for the future work, our proposed PrML can be easily extended into Kernel version to cohere with the nonlinearity in the parameter space. Besides, using SVM-style L_2-hinge loss might further improve the training efficiency [Xu *et al.*, 2016b]. Theoretical guarantees will be also investigated.

References

[Bhatia *et al.*, 2015] Kush Bhatia, Himanshu Jain, Purushottam Kar, Manik Varma, and Prateek Jain. Sparse local embeddings for extreme multi-label classification. In *Advances in Neural Information Processing Systems*, pages 730–738, 2015.

[Chiang *et al.*, 2016] Wei-Lin Chiang, Mu-Chu Lee, and Chih-Jen Lin. Parallel dual coordinate descent method for large-scale linear classification in multi-core environments. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2016.

[Hou *et al.*, 2016] Peng Hou, Xin Geng, and Min-Ling Zhang. Multi-label manifold learningn. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[Hsieh *et al.*, 2008] Cho-Jui Hsieh, Kai-Wei Chang, Chih-Jen Lin, S Sathiya Keerthi, and Sellamanickam Sundararajan. A dual coordinate descent method for large-scale linear svm. In *Proceedings of the 25th international conference on Machine learning*, pages 408–415. ACM, 2008.

[Li *et al.*, 2015] Ximing Li, Jihong Ouyang, and Xiaotang Zhou. Supervised topic models for multi-label classification. *Neurocomputing*, 149:811–819, 2015.

[Li *et al.*, 2016] Wen Li, Dengxin Dai, Mingkui Tan, Dong Xu, and Luc Van Gool. Fast algorithms for linear and kernel svm+. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2258–2266, 2016.

[Liu *et al.*, 2017] Tongliang Liu, Dacheng Tao, Mingli Song, and Stephen J Maybank. Algorithm-dependent generalization bounds for multi-task learning. *IEEE transactions on pattern analysis and machine intelligence*, 39(2):227–241, 2017.

[Pechyony and Vapnik, 2010] Dmitry Pechyony and Vladimir Vapnik. On the theory of learnining with privileged information. In *Advances in neural information processing systems*, pages 1894–1902, 2010.

[Pechyony *et al.*, 2010] Dmitry Pechyony, Rauf Izmailov, Akshay Vashist, and Vladimir Vapnik. Smo-style algorithms for learning using privileged information. In *DMIN*, pages 235–241, 2010.

[Read *et al.*, 2011] Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. Classifier chains for multi-label classification. *Machine learning*, 85(3):333–359, 2011.

[Sharmanska *et al.*, 2013] Viktoriia Sharmanska, Novi Quadrianto, and Christoph H Lampert. Learning to rank using privileged information. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 825–832, 2013.

[Sun *et al.*, 2014] Fuming Sun, Jinhui Tang, Haojie Li, Guo-Jun Qi, and Thomas S Huang. Multi-label image categorization with sparse factor representation. *Image Processing, IEEE Transactions on*, 23(3):1028–1037, 2014.

[Tai and Lin, 2012] Farbound Tai and Hsuan-Tien Lin. Multilabel classification with principal label space transformation. *Neural Computation*, 24(9):2508–2542, 2012.

[Takac *et al.*, 2015] Martin Takac, Peter Richtarik, and Nathan Srebro. Distributed mini-batch sdca. *arXiv preprint arXiv:1507.08322*, 2015.

[Tsoumakas *et al.*, 2010] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Mining multi-label data. In *Data mining and knowledge discovery handbook*, pages 667–685. Springer, 2010.

[Tsoumakas *et al.*, 2011] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. Random k-labelsets for multilabel classification. *IEEE Transactions on Knowledge and Data Engineering*, 23(7):1079–1089, 2011.

[Vapnik and Izmailov, 2015] Vladimir Vapnik and Rauf Izmailov. Learning using privileged information: Similarity control and knowledge transfer. *Journal of Machine Learning Research*, 16:2023–2049, 2015.

[Vapnik and Vashist, 2009] Vladimir Vapnik and Akshay Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5):544–557, 2009.

[Vapnik *et al.*, 2009] Vladimir Vapnik, Akshay Vashist, and Natalya Pavlovitch. Learning using hidden information (learning with teacher). In *2009 International Joint Conference on Neural Networks*, pages 3188–3195. IEEE, 2009.

[Xu *et al.*, 2015] Chang Xu, Dacheng Tao, and Chao Xu. Large-margin multi-label causal feature learning. In *AAAI*, pages 1924–1930, 2015.

[Xu *et al.*, 2016a] Chang Xu, Dacheng Tao, and Chao Xu. Robust extreme multi-label learning. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA August*, pages 13–17, 2016.

[Xu *et al.*, 2016b] Xinxing Xu, Joey Tianyi Zhou, IvorW Tsang, Zheng Qin, Rick Siow Mong Goh, and Yong Liu. Simple and efficient learning using privileged information. *arXiv preprint arXiv:1604.01518*, 2016.

[Yang *et al.*, 2016] Hao Yang, Joey Tianyi Zhou, Yu Zhang, Bin-Bin Gao, Jianxin Wu, and Jianfei Cai. Exploit bounding box annotations for multi-label object recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[Yu *et al.*, 2014] Hsiang-Fu Yu, Prateek Jain, Purushottam Kar, and Inderjit Dhillon. Large-scale multi-label learning with missing labels. In *Proceedings of The 31st International Conference on Machine Learning*, pages 593–601, 2014.

[Zhang and Wu, 2015] Min-Ling Zhang and Lei Wu. Lift: Multi-label learning with label-specific features. *IEEE transactions on pattern analysis and machine intelligence*, 37(1):107–120, 2015.

[Zhang *et al.*, 2017] Hongyang Zhang, Shan You, Zhouchen Lin, and Chao Xu. Fast compressive phase retrieval under bounded noise. In *Thirty-first AAAI Conference on Artificial Intelligence*, 2017.