Multi-Label Relationship Learning

Yu Zhang, Department of Computer Science, Hong Kong Baptist University
Dit-Yan Yeung, Department of Computer Science and Engineering, Hong Kong University of Science and Technology

Multi-label learning problems are commonly found in many applications. A characteristic shared by many multi-label learning problems is that some labels have significant correlations between them. In this paper, we propose a novel multi-label learning method, called multi-label relationship learning (MLRL), which extends the conventional support vector machine by explicitly learning and utilizing the relationships between labels. Specifically, we model the label relationships using a label covariance matrix and use it to define a new regularization term for the optimization problem. MLRL learns the model parameters and the label covariance matrix simultaneously based on a unified convex formulation. To solve the convex optimization problem, we use an alternating method in which each subproblem can be solved efficiently. The relationship between MLRL and two widely used maximum margin methods for multi-label learning is investigated. Moreover, we also propose a semi-supervised extension of MLRL, called SSMLRL, to demonstrate how to make use of unlabeled data to help learn the label covariance matrix. Through experiments conducted on some multi-label applications, we find that MLRL not only gives higher classification accuracy but also has better interpretability as revealed by the label covariance matrix.

Categories and Subject Descriptors: I.2.6 [**Artificial Intelligence**]: Learning; H.2.8 [**Database Management**]: Database Applications—*Data mining*

General Terms: Algorithms

Additional Key Words and Phrases: Multi-Label Learning, Label Relationship

1. INTRODUCTION

Different from conventional multi-class classification problems in which each input data point is associated with one and only one class label, each data point in multi-label learning problems can be associated with one or multiple labels. Multi-label learning problems are commonly found in many applications. For example, in text classification, a document may belong to several topics; in bioinformatics, a gene may perform multiple functions; in scene classification, an image may belong to several semantic classes. An important characteristic commonly found in many multi-label learning problems is that some labels have significant correlations between them. For example, a news article on finance is likely to also belong to politics, and an image is unlikely to be associated with both the ocean and urban labels simultaneously. As such, discovering the correlation between labels is a

Email: yuzhang@comp.hkbu.edu.hk, dyyeung@cse.ust.hk

crucial issue in multi-label learning.

Many multi-label learning methods have been proposed. The naïve method is to treat a multi-label problem as several separate binary classification problems. However, this scheme completely ignores the label correlation information and hence the performance of methods based on this scheme is often not very satisfactory. Some methods model multi-label problems as ranking problems [Schapire and Singer 2000; Elisseeff and Weston 2001; Fürnkranz et al. 2008] in such a way that the ranking function value of a label to which a data point belongs is larger than that of another label to which it does not belong. This approach works well for many applications. However, from the optimization perspective, modeling a classification problem as a ranking problem requires the conversion of a problem with $O(nm)$ constraints into one with $O(nm^2)$ constraints where n is the number of training data points and m is the total number of labels. Obviously this leads to an increase in problem complexity. Moreover, it is not easy to consider label correlation in ranking problems. Some multi-label learning methods are variants of conventional binary or multi-class classification methods. For example, Schapire and Singer proposed a boosting method for multi-label learning [Schapire and Singer 2000]; Clare and King extended decision trees for multi-label learning by generalizing the definition of entropy [Clare and King 2001]; Zhang and Zhou proposed a multi-label neural network by assuming that different labels share the same hidden representation corresponding to the hidden layers [Zhang and Zhou 2006]; Zhang and Zhou also proposed ML-KNN as a generalization of the k-nearest neighbor algorithm for multi-label problems [Zhang and Zhou 2007]. Moreover, Ji et al. [Ji et al. 2010] assume the model parameters for different labels share a common low-dimensional subspace and propose a regularized method to learn the shared subspace. More recently, Zhang and Zhang [Zhang and Zhang 2010] proposed a two-stage method called LEAD, which first learns the label dependency using a Bayesian network and then utilizes the learned dependency to learn multiple binary classifiers for multi-label learning. Readers are referred to [Tsoumakas and Katakis 2007; Tsoumakas et al. 2010] for a review of multi-label learning methods in the literature.

In this paper, we extend the conventional support vector machine (SVM) for multi-label learning by explicitly learning and utilizing the relationships between labels. Specifically, we model the label relationships using a label covariance matrix and use it to define a new regularization term for the objective function of the (multi-label) SVM. We call this method multi-label relationship learning (MLRL), which learns the model parameters and the label covariance matrix simultaneously based on a unified convex formulation. To solve the convex optimization problem, we use an alternating method in which each subproblem can be solved efficiently. By investigating the dual form of one subproblem which is similar to the dual form of SVM, we propose an SMO-style algorithm for solving it. To gain more insights into MLRL, we investigate the relationship between it and two widely used maximum margin multi-label learning methods, namely, conventional SVM (here we call it BSVM for multi-label learning) [Boutell et al. 2004] and RankSVM [Elisseeff and Weston 2001]. We also propose a semi-supervised extension of MLRL, called SSMLRL, which makes use of unlabeled data to help learn the label covariance matrix when the labeled data is scarce. Moreover, the label covariance matrix learned in MLRL allows us to explicitly describe the relationships between labels which may be useful for some applications as to be illustrated in our experiments. Compared with LEAD [Zhang and Zhang 2010], our method can learn the label correlation and model parameters simultaneously based on a unified and consistent framework. Moreover, learning of the label covariance matrix for characterizing label correlation is generally more efficient and effective than learning the structure of a Bayesian network.

Our MLRL method is discussed in the next section and its relationship with some related works is discussed in Section 3. Section 4 introduces the semi-supervised extension of MLRL. Section 5 reports some experimental results on several multi-label applications and Section 6 concludes the paper.

2. MULTI-LABEL RELATIONSHIP LEARNING

Suppose we are given n training data points $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$ and its label vector $\mathbf{y}_i \in \{-1, 1\}^m$. The data point $\mathbf{x}_i$ is associated with the jth label if and only if the jth element of the vector $\mathbf{y}_i$, denoted as y_i^j, is equal to 1. The predictive function for the jth label is defined as $f_j(\mathbf{x}) = \mathbf{w}_j^T \phi(\mathbf{x}) + b_j$ where $\phi(\cdot)$ denotes the feature mapping corresponding to a kernel function $k(\cdot, \cdot)$ which converts $\mathbf{x} \in \mathbb{R}^d$ to $\phi(\mathbf{x}) \in \mathbb{R}^{d'}$.

2.1 Objective Function

Recall that from a probabilistic viewpoint, SVM can be seen as obtaining the *maximum a posteriori* (MAP) solution of the following model [Kwok 1999]:

$$\mathbf{w} \sim \mathcal{N}(\mathbf{0}_{d'}, \epsilon^2 \mathbf{I}_{d'}) \tag{1}$$

$$p(y_i | \mathbf{x}_i, \mathbf{w}) \propto \exp\Big(- \big[1 - y_i(\mathbf{w}^T \phi(\mathbf{x}_i) + b)\big]_+ \Big), \ i = 1, \ldots, n, \tag{2}$$

where $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is the training set with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$, $\mathbf{0}_d$ is the $d \times 1$ zero vector, $\mathbf{I}_d$ is the $d \times d$ identity matrix, $\mathcal{N}(\mathbf{m}, \mathbf{\Sigma})$ denotes the multivariate (or univariate) normal distribution with mean $\mathbf{m}$ and covariance matrix (or variance) $\mathbf{\Sigma}$, and $[u]_+ \overset{\text{def}}{=} \max(0, u)$. Eq. (1) defines the prior on the model parameter $\mathbf{w}$ and Eq. (2) defines the data likelihood.

Similar to SVM, we propose a similar probabilistic model for multi-label learning:

$$\mathbf{W} \sim \mathcal{MN}_{d' \times m}(\mathbf{W} | \mathbf{0}_{d' \times m}, \mathbf{I}_{d'} \otimes \mathbf{\Omega})$$
$$p(\mathbf{y}_i | \mathbf{x}_i, \mathbf{W}) \propto \prod_{j=1}^m \exp\Big(- \big[1 - y_i^j(\mathbf{w}_j^T \phi(\mathbf{x}_i) + b_j)\big]_+ \Big), \tag{3}$$

where $\mathbf{W} = (\mathbf{w}_1, \ldots, \mathbf{w}_m)$, $\mathcal{MN}_{d \times m}(\mathbf{M}, \mathbf{A} \otimes \mathbf{B})$ denotes a matrix-variate normal distribution[1] [Gupta and Nagar 2000] with mean $\mathbf{M} \in \mathbb{R}^{d \times m}$, row covariance matrix $\mathbf{A} \in \mathbb{R}^{d \times d}$ and column covariance matrix $\mathbf{B} \in \mathbb{R}^{m \times m}$, and $\mathbf{0}_{d \times m}$ denotes the $d \times m$ zero matrix. The prior on $\mathbf{W}$ is to model the structure of $\mathbf{W}$. More specifically, the row covariance matrix $\mathbf{I}_d$ models the relationships between features and the column covariance matrix $\mathbf{\Omega}$ models the relationships between different $\mathbf{w}_i$'s. In other words, $\mathbf{\Omega}$ models the relationships between labels. When $\mathbf{\Omega} \propto \mathbf{I}_m$, this model will decompose into m binary SVM models. For most applications, $\mathbf{\Omega}$ is not known *a priori* and so we seek to estimate it from data automatically.

[1] The probability density function of a matrix-variate normal distribution is defined as $p(\mathbf{X}|\mathbf{M}, \mathbf{A}, \mathbf{B}) = \frac{\exp\left(-\frac{1}{2}\text{tr}\left(\mathbf{A}^{-1}(\mathbf{X}-\mathbf{M})\mathbf{B}^{-1}(\mathbf{X}-\mathbf{M})^T\right)\right)}{(2\pi)^{md/2}|\mathbf{A}|^{m/2}|\mathbf{B}|^{d/2}}$.

Then the optimization problem for the MAP solution is formulated as follows:

$$\min_{\mathbf{W},\mathbf{b},\mathbf{\Omega}\succ 0} \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} L(y_i^j, \mathbf{w}_j^T\phi(\mathbf{x}_i)+b_j) + \frac{1}{2n}\mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T) + \frac{d'}{2n}\ln|\mathbf{\Omega}|, \qquad (4)$$

where $\mathrm{tr}(\cdot)$ denotes the trace of a square matrix, $L(\cdot,\cdot)$ denotes the hinge loss, $\mathbf{\Omega}$ is the positive definite (PD) label covariance matrix, $|\cdot|$ denotes the determinant of a square matrix, $\mathbf{b} = (b_1,\ldots,b_m)^T$, and $\mathbf{A} \succ 0$ means that the matrix $\mathbf{A}$ is PD. The first term in (4) measures the empirical loss on the training data and the second term penalizes the complexity of $\mathbf{W}$ and measures the relationships between all labels based on $\mathbf{W}$ and $\mathbf{\Omega}$, where $\mathbf{\Omega}$ is a label covariance matrix which describes the relationships between labels. Since the third term $\ln|\mathbf{\Omega}|$ in problem (4) is a non-convex function with respect to $\mathbf{\Omega}$, we upper bound this term with a convex function $\mathrm{tr}(\mathbf{\Omega})$ due to the fact that $\ln|\mathbf{\Omega}| \le \mathrm{tr}(\mathbf{\Omega}) - m$.[2] Then the optimization problem becomes

$$\min_{\mathbf{W},\mathbf{b},\mathbf{\Omega}\succeq 0} \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} L(y_i^j, \mathbf{w}_j^T\phi(\mathbf{x}_i)+b_j) + \frac{1}{2n}\mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T) + \frac{d'}{2n}\mathrm{tr}(\mathbf{\Omega}). \qquad (5)$$

Moreover, for a feature mapping $\phi(\cdot)$ corresponding to a kernel, the dimensionality d' may be infinite which makes problem (5) infeasible. So by using the method of Lagrange multipliers, we can get a problem which is equivalent to problem (5):

$$\begin{aligned}\min_{\mathbf{W},\mathbf{b},\mathbf{\Omega}} \quad & \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} L(y_i^j, \mathbf{w}_j^T\phi(\mathbf{x}_i)+b_j) + \frac{\lambda}{2}\mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T)\\ \text{s.t.} \quad & \mathbf{\Omega} \succ 0,\\ & \mathrm{tr}(\mathbf{\Omega}) = 1, \qquad (6)\end{aligned}$$

where λ is a regularization parameter.

2.2 Optimization Procedure

We first prove the convexity of problem (6) with respect to all variables.

THEOREM 1. *Problem (6) is convex with respect to* $\mathbf{W}$*,* $\mathbf{b}$ *and* $\mathbf{\Omega}$*.*

Proof:
It is easy to see that the first term in the objective function of problem (6) is convex with respect to all variables and the constraints in (6) are also convex. It has been proved in [Zhang and Yeung 2010] that the second term in the objective function is a convex function with respect to $\mathbf{W}$, $\mathbf{b}$ and $\mathbf{\Omega}$. So the objective function and the constraints in problem (6) are convex with respect to all variables and hence problem (6) is jointly convex. □

Problem (6) is a semidefinite programming (SDP) problem due to the first constraint. We will present below an efficient algorithm for solving it.

Even though the optimization problem (6) is convex with respect to $\mathbf{W}$, $\mathbf{b}$ and $\mathbf{\Omega}$ jointly, it is not easy to optimize the objective function with respect to all the variables simultaneously. Here we propose an alternating method to solve the problem more efficiently. Specifically, we first optimize the objective function with respect to $\mathbf{W}$ and $\mathbf{b}$ when $\mathbf{\Omega}$

[2] Due to the fact that $\ln x \le x - 1$ for a positive scalar x, we have $\ln|\mathbf{\Omega}| = \sum_{i=1}^m \ln\eta_i \le \sum_{i=1}^m(\eta_i - 1) = \mathrm{tr}(\mathbf{\Omega}) - m$ where η_i is the ith largest eigenvalue of $\mathbf{\Omega}$.

is fixed, and then optimize it with respect to $\mathbf{\Omega}$ when $\mathbf{W}$ and $\mathbf{b}$ are fixed. This procedure is repeated until convergence. In what follows, we will present the two subproblems separately.

Optimizing w.r.t. $\mathbf{W}$ and $\mathbf{b}$ when $\mathbf{\Omega}$ is fixed. When $\mathbf{\Omega}$ is given and fixed, the optimization problem for finding $\mathbf{W}$ and $\mathbf{b}$ is an unconstrained convex optimization problem provided that the loss function is convex. The optimization problem can be stated as:

$$\min_{\mathbf{W},\mathbf{b}} \quad \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m} L(y_i^j, \mathbf{w}_j^T\phi(\mathbf{x}_i)+b_j) + \frac{\lambda}{2}\mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T). \tag{7}$$

We reformulate the optimization problem into a dual form by first expressing problem (7) as a constrained optimization problem:

$$\begin{aligned}\min_{\mathbf{W},\mathbf{b},\{\varepsilon_i^j\}} & \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\varepsilon_i^j + \frac{\lambda}{2}\mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T)\\ \text{s.t. } & y_i^j(\mathbf{w}_j^T\phi(\mathbf{x}_i)+b_j) \geq 1-\varepsilon_i^j,\ \varepsilon_i^j \geq 0.\end{aligned} \tag{8}$$

The Lagrangian of problem (8) is given by

$$\begin{aligned}G =& \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{m}\varepsilon_i^j + \frac{\lambda}{2}\mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T) + \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i^j\left[1-y_i^j(\mathbf{w}_j^T\phi(\mathbf{x}_i)+b_j)-\varepsilon_i^j\right]\\ &-\sum_{i=1}^{n}\sum_{j=1}^{m}\beta_i^j\varepsilon_i^j,\end{aligned}$$

where $\alpha_i^j, \beta_i^j \geq 0$. We calculate the gradients of G with respect to $\mathbf{W}$, b_j and ε_i^j and set them to 0 to obtain

$$\begin{aligned}\frac{\partial G}{\partial \mathbf{W}} &= \lambda\mathbf{W}\mathbf{\Omega}^{-1} - \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i^j y_i^j\phi(\mathbf{x}_i)\mathbf{e}_j^T = 0 \qquad (9)\\ \frac{\partial G}{\partial b_j} &= -\sum_{i=1}^{n}\alpha_i^j y_i^j = 0\\ \frac{\partial G}{\partial \varepsilon_i^j} &= \frac{1}{n} - \alpha_i^j - \beta_i^j = 0,\end{aligned}$$

where $\mathbf{e}_i$ is the ith column vector of $\mathbf{I}_m$. Plugging the above equations into the Lagrangian, we obtain the following dual form:

$$\begin{aligned}\min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T\tilde{\mathbf{K}}_{ML}\boldsymbol{\alpha} - \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i^j\\ \text{s.t.} \quad & \sum_{i=1}^{n}\alpha_i^j y_i^j = 0,\\ & 0 \leq \alpha_i^j \leq \frac{1}{n},\ \forall i,j\end{aligned} \tag{10}$$

where $\boldsymbol{\alpha} = (\alpha_1^1,\ldots,\alpha_n^1,\ldots,\alpha_1^m,\ldots,\alpha_n^m)^T$, $\mathbf{y} = (y_1^1,\ldots,y_n^1,\ldots,y_1^m,\ldots,y_n^m)^T$, $\mathbf{K}$ is the kernel matrix on all data points, $\mathbf{K}_{ML} = \frac{1}{\lambda}\mathbf{\Omega} \otimes \mathbf{K}$, $\otimes$ denotes the Kronecker product, $\tilde{\mathbf{K}}_{ML} = (\mathbf{K}_{ML} \odot \mathbf{y}\mathbf{y}^T)$, and $\odot$ denotes the matrix elementwise product operator.

From the formulation of $\mathbf{K}_{ML}$, we can define the multi-label kernel as

$$k_{ML}((\mathbf{x}_i, j), (\mathbf{x}_p, q)) = \frac{1}{\lambda}\Omega_{jq}k(\mathbf{x}_i, \mathbf{x}_p)$$

which describes the similarity betweem $\mathbf{x}_i$ and $\mathbf{x}_p$ based on Ω_{jq} as the (j, q)th element of $\mathbf{\Omega}$ when they hold the jth and qth labels, respectively. From this formulation, we can see how $\mathbf{\Omega}$ plays the role in the definition of data similarity. The multi-label kernel here is different from the conventional kernel used in previous multi-label methods. Our multi-label kernel, which incorporates the label covariance into the definition of kernel function, describes the similarity between two data points when they are associated with some labels. From the definition of multi-label kernel, we can see that when two labels are more correlated, the multi-label kernel function will have a larger value due to the larger value of Ω_{jq}.

Note that the dual problem (10) is very similar to that of SVM except the first constraint in problem (10). For SVM, there is only one constraint, but here there are m constraints with one constraint for each label. Problem (10) is a quadratic programming (QP) problem which is computationally demanding when solved directly, so we develop an SMO-style algorithm similar to that in [Fan et al. 2005] to solve it. We defer the discussion of the optimization method for problem (10) to Appendix A.

Optimizing w.r.t. $\mathbf{\Omega}$ when $\mathbf{W}$ and $\mathbf{b}$ are fixed. When $\mathbf{W}$ and $\mathbf{b}$ are fixed, the optimization problem for finding $\mathbf{\Omega}$ becomes

$$\begin{aligned} \min_{\mathbf{\Omega}} \quad & \mathrm{tr}(\mathbf{\Omega}^{-1}\mathbf{W}^T\mathbf{W}) \\ \text{s.t.} \quad & \mathbf{\Omega} \succ 0 \\ & \mathrm{tr}(\mathbf{\Omega}) = 1. \end{aligned} \tag{11}$$

We lower bound the objective as

$$\begin{aligned} \mathrm{tr}(\mathbf{\Omega}^{-1}\mathbf{A}) &= \mathrm{tr}(\mathbf{\Omega}^{-1}\mathbf{A})\mathrm{tr}(\mathbf{\Omega}) \\ &= \mathrm{tr}\Big((\mathbf{\Omega}^{-\frac{1}{2}}\mathbf{A}^{\frac{1}{2}})(\mathbf{A}^{\frac{1}{2}}\mathbf{\Omega}^{-\frac{1}{2}})\Big)\mathrm{tr}(\mathbf{\Omega}^{\frac{1}{2}}\mathbf{\Omega}^{\frac{1}{2}}) \\ &\geq (\mathrm{tr}(\mathbf{\Omega}^{-\frac{1}{2}}\mathbf{A}^{\frac{1}{2}}\mathbf{\Omega}^{\frac{1}{2}}))^2 = (\mathrm{tr}(\mathbf{A}^{\frac{1}{2}}))^2, \end{aligned}$$

where $\mathbf{A} = \mathbf{W}^T\mathbf{W}$. The first equality holds because of the last constraint in problem (11) and the last inequality holds because of the Cauchy-Schwarz inequality for the Frobenius norm. Moreover, $\mathrm{tr}(\mathbf{\Omega}^{-1}\mathbf{A})$ attains its minimum value $(\mathrm{tr}(\mathbf{A}^{\frac{1}{2}}))^2$ if and only if $\mathbf{\Omega}^{-\frac{1}{2}}\mathbf{A}^{\frac{1}{2}} = a\mathbf{\Omega}^{\frac{1}{2}}$ for some constant a and $\mathrm{tr}(\mathbf{\Omega}) = 1$. So we can get the analytical solution $\mathbf{\Omega} = \frac{(\mathbf{W}^T\mathbf{W})^{\frac{1}{2}}}{\mathrm{tr}((\mathbf{W}^T\mathbf{W})^{\frac{1}{2}})}$. Here we compute $\mathbf{W}^T\mathbf{W}$ first. From Eq. (9), we can get

$$\mathbf{W} = \frac{1}{\lambda}\sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i^j y_i^j \phi(\mathbf{x}_i)\mathbf{e}_j^T\mathbf{\Omega}.$$

Then we can calculate $\mathbf{W}^T\mathbf{W}$ as

$$\mathbf{W}^T\mathbf{W} = \frac{1}{\lambda^2}\sum_{i,j}\sum_{p,q}\alpha_i^j\alpha_p^q y_i^j y_p^q k(\mathbf{x}_i, \mathbf{x}_p)\mathbf{\Omega}\mathbf{e}_j\mathbf{e}_q^T\mathbf{\Omega}.$$

When the number of tasks is less than the number of feature dimensions which holds when using kernels such as the RBF kernel, $\mathbf{\Omega}$ is a PD matrix. Otherwise we can regularize it by adding a scaled identity matrix to ensure that it is PD.

We set the initial value of $\mathbf{\Omega}$ to $\frac{1}{m}\mathbf{I}_m$ which corresponds to the assumption that all labels are unrelated initially. After learning the optimal values of $\mathbf{W}$, $\mathbf{b}$ and $\mathbf{\Omega}$, we can make prediction for any new data point. Given a test data point $\mathbf{x}_\star$, the predictive output $y_\star^i$ for the ith label is given by $y_\star^i = \mathrm{sign}(t_\star^i)$ where $\mathrm{sign}(\cdot)$ is the sign function, and $t_\star^i = \sum_{p=1}^n \sum_{q=1}^m \alpha_p^q y_p^q k_{ML}((\mathbf{x}_p, q), (\mathbf{x}_\star, i)) + b_i$.

2.3 Implementation Issues

In problem (10), it appears that we need to store a large matrix of size $nm \times nm$, but in fact we can exploit the structure of $\mathbf{K}_{ML}$ to avoid this. Since $\mathbf{K}_{ML} = \frac{1}{\lambda}\mathbf{\Omega} \otimes \mathbf{K}$, we only need to store $\mathbf{K}$ and $\mathbf{\Omega}$ which only cost $O(n^2 + m^2)$ space. Moreover, in our SMO algorithm, we do not need to store the entire $\mathbf{K}$ but can calculate the kernel function value on demand.

The Kronecker product $\otimes$ usually converts two small matrices into a very large matrix which can be verified by comparing the sizes of $\mathbf{K}$, $\mathbf{\Omega}$ and $\mathbf{K}_{ML}$. However, we can make use of some properties of the Kronecker product to reduce the space requirement, as illustrated here. For example, if we want to calculate the objective function value of problem (10), the first term can be simplified as

$$\begin{aligned}
\boldsymbol{\alpha}^T \tilde{\mathbf{K}}_{ML} \boldsymbol{\alpha} &= \boldsymbol{\alpha}^T (\mathbf{K}_{ML} \odot \mathbf{y}\mathbf{y}^T)\boldsymbol{\alpha} \\
&= \boldsymbol{\alpha}^T \mathrm{diag}(\mathbf{y}) \mathbf{K}_{ML} \mathrm{diag}(\mathbf{y}) \boldsymbol{\alpha} \\
&= (\boldsymbol{\alpha} \odot \mathbf{y})^T \mathbf{K}_{ML} (\boldsymbol{\alpha} \odot \mathbf{y}) \\
&= \mathrm{vec}(\mathbf{M} \odot \mathbf{Y})^T (\tilde{\mathbf{\Omega}} \otimes \mathbf{K}) \mathrm{vec}(\mathbf{M} \odot \mathbf{Y}) \\
&= \mathrm{tr}\Big((\mathbf{M} \odot \mathbf{Y})^T \mathbf{K} (\mathbf{M} \odot \mathbf{Y}) \tilde{\mathbf{\Omega}} \Big),
\end{aligned}$$

where $\tilde{\mathbf{\Omega}} = \frac{1}{\lambda}\mathbf{\Omega}$, $\mathrm{vec}(\cdot)$ denotes the operator which converts a matrix to a vector in the columnwise order, $\mathbf{M}$ is an $n \times m$ matrix such that $\mathrm{vec}(\mathbf{M}) = \boldsymbol{\alpha}$, $\mathbf{Y}$ is an $n \times m$ matrix such that $\mathrm{vec}(\mathbf{Y}) = \mathbf{y}$, and $\mathrm{diag}(\cdot)$ denotes the operator which converts a vector to a diagonal matrix. The last equality holds due to a property of the operators $\mathrm{vec}(\cdot)$ and $\otimes$, namely, $\mathrm{tr}(\mathbf{ABCD}) = \mathrm{vec}(\mathbf{A}^T)^T (\mathbf{D}^T \otimes \mathbf{B}) \mathrm{vec}(\mathbf{C})$ for any $\mathbf{A} \in \mathbb{R}^{a \times b}$, $\mathbf{B} \in \mathbb{R}^{b \times c}$, $\mathbf{C} \in \mathbb{R}^{c \times d}$ and $\mathbf{D} \in \mathbb{R}^{d \times a}$.

In the beginning of our SMO algorithm, we need to calculate $f_j^i = \boldsymbol{\alpha}^T (\mathbf{k}_j^i \odot \mathbf{y}) - y_j^i$ where $\mathbf{k}_j^i$ is a column of $\mathbf{K}_{ML}$. Recognizing that $\mathbf{k}_j^i = \boldsymbol{\omega}_i \otimes \tilde{\mathbf{k}}_j$ where $\boldsymbol{\omega}_i$ is the ith column of $\tilde{\mathbf{\Omega}}$ and $\tilde{\mathbf{k}}_j$ is the jth column of $\mathbf{K}$, we can get

$$\begin{aligned}
\boldsymbol{\alpha}^T (\mathbf{k}_j^i \odot \mathbf{y}) &= (\boldsymbol{\alpha} \odot \mathbf{y})^T \mathbf{k}_j^i \\
&= (\mathbf{k}_j^i)^T (\boldsymbol{\alpha} \odot \mathbf{y}) \\
&= (\boldsymbol{\omega}_i \otimes \tilde{\mathbf{k}}_j)^T (\boldsymbol{\alpha} \odot \mathbf{y}) \\
&= (\boldsymbol{\omega}_i^T \otimes \tilde{\mathbf{k}}_j^T) \mathrm{vec}(\mathbf{M} \odot \mathbf{Y}) \\
&= \mathrm{vec}(\tilde{\mathbf{k}}_j^T (\mathbf{M} \odot \mathbf{Y}) \boldsymbol{\omega}_i) \\
&= \tilde{\mathbf{k}}_j^T (\mathbf{M} \odot \mathbf{Y}) \boldsymbol{\omega}_i.
\end{aligned}$$

Here we use the fact that $\mathrm{vec}(\mathbf{ABC}) = (\mathbf{C}^T \otimes \mathbf{A}) \mathrm{vec}(\mathbf{B})$ for any matrices $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{C}$ of proper sizes. Similarly, when a test data point $\mathbf{x}_\star$ is given, its output can be calculated as

$$\begin{aligned}\mathbf{y}_\star &= (\tilde{\mathbf{\Omega}} \otimes \mathbf{k}_\star)(\boldsymbol{\alpha} \odot \mathbf{y}) + \mathbf{b}\\ &= (\tilde{\mathbf{\Omega}} \otimes \mathbf{k}_\star)\mathrm{vec}(\mathbf{M} \odot \mathbf{Y}) + \mathbf{b}\\ &= \mathrm{vec}(\mathbf{k}_\star(\mathbf{M} \odot \mathbf{Y})\tilde{\mathbf{\Omega}}) + \mathbf{b}\\ &= \tilde{\mathbf{\Omega}}(\mathbf{M} \odot \mathbf{Y})^T\mathbf{k}_\star^T + \mathbf{b},\end{aligned}$$

where $\mathbf{k}_\star = (k(\mathbf{x}_\star, \mathbf{x}_1), \ldots, k(\mathbf{x}_\star, \mathbf{x}_n))$.

3. RELATIONSHIP WITH PREVIOUS METHODS

Two widely used maximum margin methods for multi-label learning are BSVM [Boutell et al. 2004] and RankSVM [Elisseeff and Weston 2001]. BSVM decomposes a multi-label problem into a set of binary classification problems and RankSVM formulates a multi-label problem as a ranking problem and uses a ranking loss for optimization. In the following, we will discuss the relationship between these methods and MLRL.

Using the notations in Section 2, the optimization problem of BSVM for the jth label can be defined as follows:

$$\min_{\mathbf{w}_j, b_j} \quad \frac{1}{n}\sum_{i=1}^{n} L(y_i^j, \mathbf{w}_j^T\phi(\mathbf{x}_i) + b_j) + \frac{\lambda}{2}\mathbf{w}_j^T\mathbf{w}_j.$$

Its dual form is

$$\begin{aligned}\min_{\boldsymbol{\alpha}_j} \quad & \frac{1}{2\lambda}\boldsymbol{\alpha}_j^T(\mathbf{K} \odot \mathbf{y}_j\mathbf{y}_j^T)\boldsymbol{\alpha}_j - \sum_{i=1}^{n}\alpha_i^j\\ \text{s.t.} \quad & \sum_{i=1}^{n}\alpha_i^j y_i^j = 0,\ 0 \le \alpha_i^j \le \frac{1}{n},\end{aligned}$$

where $\boldsymbol{\alpha}_j = (\alpha_1^j, \ldots, \alpha_n^j)^T$ and $\mathbf{y}_j = (y_1^j, \ldots, y_n^j)^T$. Combining the dual optimization problems for all labels, the overall dual optimization problem is

$$\begin{aligned}\min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T(\tilde{\mathbf{K}} \odot \mathbf{y}\mathbf{y}^T)\boldsymbol{\alpha} - \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i^j\\ \text{s.t.} \quad & \sum_{i=1}^{n}\alpha_i^j y_i^j = 0, \forall j\\ & 0 \le \alpha_i^j \le \frac{1}{n}\ \forall i, j,\end{aligned} \tag{12}$$

where $\tilde{\mathbf{K}} = \frac{1}{\lambda}\mathbf{I}_m \otimes \mathbf{K}$. We can see that the dual problem (12) of BSVM is almost identical to that in (10) except for the difference between $\mathbf{K}_{ML}$ and $\tilde{\mathbf{K}}$. Recall that

$$\mathbf{K}_{ML} = \frac{1}{\lambda}\mathbf{\Omega} \otimes \mathbf{K}.$$

So when $\mathbf{\Omega} = \mathbf{I}_m$, $\mathbf{K}_{ML}$ degenerates to $\tilde{\mathbf{K}}$. From this analysis, we can understand the role of $\mathbf{\Omega}$ in describing the relationships between labels.

For RankSVM, the optimization problem is formulated as

$$\min_{\mathbf{W},\mathbf{b}} \frac{1}{n}\sum_{i=1}^{n}\sum_{j,k=1}^{m} I(y_i^j \neq y_i^k) L\Big(\frac{y_i^j - y_i^k}{2}\big((\mathbf{w}_j - \mathbf{w}_k)^T\phi(\mathbf{x}_i) + b_j - b_k\big)\Big) + \frac{\lambda}{2}\mathrm{tr}(\mathbf{W}\mathbf{W}^T),$$

where $I(z)$ is an indicator function which outputs 1 when z is true and 0 otherwise. Then the dual form is given by

$$\begin{aligned} \min_{\boldsymbol{\alpha},\{\beta_i^{j,k}\}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T(\tilde{\mathbf{K}} \odot \mathbf{y}\mathbf{y}^T)\boldsymbol{\alpha} - \sum_{i=1}^{n}\sum_{j=1}^{m}\alpha_i^j \\ \text{s.t.} \quad & \sum_{i=1}^{n}\alpha_i^j y_i^j = 0,\ \alpha_i^j = \sum_{k=1}^{m} I(y_i^j \neq y_i^k)\beta_i^{j,k} \\ & 0 \leq \beta_i^{j,k} \leq \frac{1}{n},\ \beta_i^{j,k} = \beta_i^{k,j}. \end{aligned} \tag{13}$$

We can find the objective function for the dual form of RankSVM is the same as that of BSVM but the labels are related according to the constraints on α_i^j, but in BSVM different labels are independent. In MLRL, the labels are related via the multi-label kernel matrix $\mathbf{K}_{ML}$. Hence MLRL and RankSVM work in different ways in exploiting the relatedness of different labels. Moreover, since modeling a classification problem as a ranking problem requires using $O(m^2)$ constraints (constraints on α_i^j and $\beta_i^{j,k}$ for $1 \leq j,k \leq m$) for each data point, MLRL with only $O(m)$ constraints (constraints on α_i^j for $1 \leq j \leq m$) for each data point has a lower complexity than RankSVM from the optimization perspective.

Some methods for multi-label learning also capture the label correlations, e.g., [Bucak et al. 2009] and [Zhang and Zhang 2010]. The method in [Bucak et al. 2009] formulates the multi-label problem as a ranking problem in a way similar to RankSVM and hence the label correlations are captured implicitly in the ranking function. The LEAD method proposed in [Zhang and Zhang 2010] is a two-stage algorithm which first learns the label dependency by using a Bayesian network and then performs classification. Different from the LEAD method, our method learns the label correlations and the model parameters simultaneously. Moreover, there are some other related works in other research areas. For example, in [Argyriou et al. 2008; Argyriou et al. 2008], a multi-task feature learning (MTFL) method was proposed to utilize the trace norm as a regularizer. Different from our method which learns the label correlations, the MTFL method focuses on learning feature correlation. The trace norm regularization is also used in maximum margin matrix factorization [Srebro et al. 2004] for collaborative filtering. A recent study on learning an output kernel in [Dinuzzo et al. 2011; Dinuzzo and Fukumizu 2011] has the same objective as our method. One advantage of our method over [Dinuzzo et al. 2011; Dinuzzo and Fukumizu 2011] is that our method is jointly convex which may bring some computational benefit.

4. SEMI-SUPERVISED EXTENSION

In many real-world applications, auxiliary sources of data are available in addition to labeled data. For example, semi-supervised learning [Chapelle et al. 2006] is an active research subarea which utilizes unlabeled data available to enhance the learning accuracy. In the context of multi-label learning, some attempts have also been made in utilizing unlabeled data [Liu et al. 2006; Chen et al. 2008; Zha et al. 2009]. In this section, we discuss how to extend MLRL to the semi-supervised setting.

Suppose we are given l labeled data points $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{l}$ with $\mathbf{x}_i \in \mathbb{R}^d$ and its corresponding label vector $\mathbf{y}_i \in \{-1, 1\}^m$, as well as u unlabeled data points $\{\mathbf{x}_i\}_{i=1+1}^{l+u}$. So there are $n = l + u$ data points in the training set. The predictive function for the ith label is defined as $f_i(\mathbf{x}) = \mathbf{w}_i^T \phi(\mathbf{x}) + b_i$.

4.1 Objective Function

Unlike existing graph-based semi-supervised multi-label learning methods which capture the geometric information contained in the unlabeled data, our method utilizes unlabeled data to help estimate the label covariance matrix $\mathbf{\Omega}$. Similar to the smoothness assumption commonly used in semi-supervised learning which assumes that the decision function value varies smoothly on the data manifold, here we assume that the decision function values for different labels are smooth with respect to the label relationships. Specifically, if two labels are highly positively correlated, the decision function values of these two labels on each data point will be similar; if two labels are highly negatively correlated, the decision function values of these two labels on each data point will be dissimilar; for two independent labels, the decision function values will be independent.

Based on the above assumption, we propose the following formulation for the semi-supervised multi-label learning problem:

$$\begin{aligned} \min_{\mathbf{W},\mathbf{b},\mathbf{\Omega},\mathbf{F}} \ & \frac{1}{l}\sum_{i=1}^{l}\sum_{j=1}^{m}[1 - y_i^j(\mathbf{w}_j^T\phi(\mathbf{x}_i) + b_j)]_+ + \frac{\lambda_1}{2}\mathrm{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T) + \frac{\lambda_2}{2}\mathrm{tr}(\mathbf{F}\mathbf{\Omega}^{-1}\mathbf{F}^T) \\ \text{s.t. } & \mathbf{F} = \phi(\mathbf{X}_u)^T\mathbf{W} \\ & \mathbf{\Omega} \succ 0 \\ & \mathrm{tr}(\mathbf{\Omega}) = 1, \end{aligned} \tag{14}$$

where λ_1 and λ_2 are regularization parameters, $\phi(\mathbf{X}_u) = \Big(\phi(\mathbf{x}_{l+1}), \ldots, \phi(\mathbf{x}_n)\Big)$, and the (i, j)th element of $\mathbf{F}$, denoted by f_i^j, is the function value of the ith unlabeled data point, e.g., $f_i^j = \mathbf{w}_j^T\phi(\mathbf{x}_{l+i})$.[3] The first term in (14) measures the empirical loss on the training data, the second term measures the relationships between all labels based on $\mathbf{W}$, and the third term uses unlabeled data to help estimate $\mathbf{\Omega}$ which reflects the rationale of our assumption.

To the best of our knowledge, all existing semi-supervised multi-label learning methods [Liu et al. 2006; Chen et al. 2008; Zha et al. 2009] are *tranductive methods* and so they can only make predictions for the data points in the training set which contains both labeled and unlabeled data. So making predictions for unseen test data points is not easy using the existing methods. However, our method is an *inductive method* which can easily make predictions for unseen test data points.

4.2 Optimization Procedure

We first discuss the convexity of problem (14) with respect to all variables.

THEOREM 2. *Problem (14) is convex with respect to* $\mathbf{W}$, $\mathbf{b}$, $\mathbf{\Omega}$ *and* $\mathbf{F}$.

[3] Similar to the manifold regularization method in semi-supervised learning, the offsets $\{b_j\}$ are not included in the regularization term.

The proof of Theorem 2 is very similar to that of Theorem 1 and hence we omit it here for brevity.

Here we also use an alternating method to solve problem (14) more efficiently.

Optimizing w.r.t. $\mathbf{W}$, $\mathbf{b}$ and $\mathbf{F}$ when $\mathbf{\Omega}$ is fixed. When $\mathbf{\Omega}$ is fixed, we formulate problem (14) as

$$\begin{aligned}\min_{\mathbf{W},\mathbf{b},\{\varepsilon_i^j\},\mathbf{F}} \ & \frac{1}{l}\sum_{i=1}^{l}\sum_{j=1}^{m}\varepsilon_i^j + \frac{\lambda_1}{2}\text{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T) + \frac{\lambda_2}{2}\text{tr}(\mathbf{F}\mathbf{\Omega}^{-1}\mathbf{F}^T)\\ \text{s.t.} \ & y_i^j(\mathbf{w}_j^T\phi(\mathbf{x}_i)+b_j) \geq 1-\varepsilon_i^j \quad \forall i,j\\ & \varepsilon_i^j \geq 0 \quad \forall i,j\\ & \mathbf{F} = \phi(\mathbf{X}_u)^T\mathbf{W}. \end{aligned} \tag{15}$$

The Lagrangian of problem (15) is given by

$$\begin{aligned} G = & \frac{1}{l}\sum_{i=1}^{l}\sum_{j=1}^{m}\varepsilon_i^j + \frac{\lambda_1}{2}\text{tr}(\mathbf{W}\mathbf{\Omega}^{-1}\mathbf{W}^T) + \frac{\lambda_2}{2}\text{tr}(\mathbf{F}\mathbf{\Omega}^{-1}\mathbf{F}^T) - \sum_{i=1}^{l}\sum_{j=1}^{m}\beta_i^j\varepsilon_i^j\\ & + \sum_{i=1}^{l}\sum_{j=1}^{m}\alpha_i^j\Big[1-y_i^j(\mathbf{w}_j^T\phi(\mathbf{x}_i)+b_j)-\varepsilon_i^j\Big] + \text{tr}\Big(\mathbf{\Gamma}^T\big(\mathbf{F}-\phi(\mathbf{X}_u)^T\mathbf{W}\big)\Big), \end{aligned} \tag{16}$$

where $\alpha_i^j \geq 0$, $\beta_i^j \geq 0$ and $\mathbf{\Gamma} \in \mathbb{R}^{u\times m}$. We calculate the gradients of G with respect to $\mathbf{W}$, b_j, ε_i^j and $\mathbf{F}$ and set them to 0 to obtain

$$\frac{\partial G}{\partial \mathbf{W}} = \lambda_1\mathbf{W}\mathbf{\Omega}^{-1} - \sum_{i=1}^{l}\sum_{j=1}^{m}\alpha_i^j y_i^j\phi(\mathbf{x}_i)\mathbf{e}_j^T - \sum_{i=1}^{u}\sum_{j=1}^{m}\gamma_i^j\phi(\mathbf{x}_{l+i})\mathbf{e}_j^T = 0 \tag{17}$$

$$\begin{aligned}\frac{\partial G}{\partial b_j} &= -\sum_{i=1}^{l}\alpha_i^j y_i^j = 0\\ \frac{\partial G}{\partial \varepsilon_i^j} &= \frac{1}{l} - \alpha_i^j - \beta_i^j = 0\\ \frac{\partial G}{\partial \mathbf{F}} &= \lambda_2\mathbf{F}\mathbf{\Omega}^{-1} + \mathbf{\Gamma} = 0, \end{aligned}$$

where $\mathbf{I}_m$ denotes the $m\times m$ identity matrix and γ_i^j is the (i,j)th element of $\mathbf{\Gamma}$. Plugging the above equations into the Lagrangian, we obtain the following dual form:

$$\begin{aligned}\min_{\mathbf{\Theta},\mathbf{\Gamma}} \quad & \frac{1}{2\lambda_1}\text{tr}\bigg(\bigg(\tilde{\mathbf{Y}}\odot\begin{pmatrix}\mathbf{\Theta}\\ \mathbf{\Gamma}\end{pmatrix}\bigg)^T\mathbf{K}\bigg(\tilde{\mathbf{Y}}\odot\begin{pmatrix}\mathbf{\Theta}\\ \mathbf{\Gamma}\end{pmatrix}\bigg)\mathbf{\Omega}\bigg) + \frac{1}{2\lambda_2}\text{tr}(\mathbf{\Gamma}\mathbf{\Omega}\mathbf{\Gamma}^T) - \sum_{i=1}^{l}\sum_{j=1}^{m}\alpha_i^j\\ \text{s.t.} \quad & \sum_{i=1}^{l}\alpha_i^j y_i^j = 0, \forall j\\ & 0 \leq \alpha_i^j \leq \frac{1}{l}, \forall i,j, \end{aligned} \tag{18}$$

where $\mathbf{\Theta}$ is an $l \times m$ matrix with α_i^j as the (i, j)th element, $\mathbf{Y}$ is an $l \times m$ matrix with y_i^j as the (i, j)th element, $\mathbf{1}_{p\times q}$ denotes a $p \times q$ matrix of all one's, $\tilde{\mathbf{Y}} = (\mathbf{Y}^T, \mathbf{1}_{m\times u})^T$, and $\mathbf{K}$ is the kernel matrix on all data points including both labeled and unlabeled ones.

We have the following theorem on the convexity of problem (18). The proof is given in Appendix B.

THEOREM 3. *Problem (18) is a convex problem with respect to vec*$\Big(\begin{pmatrix} \mathbf{\Theta} \\ \mathbf{\Gamma} \end{pmatrix}\Big)$.

Even though problem (18) is a convex QP problem, the computational complexity is very high if the number of data points n is very large. In what follows, we will show how to speed up this problem.

Instead of directly solving problem (18), we partition $\mathbf{K}$ as $\mathbf{K} = \begin{pmatrix} \mathbf{K}_{ll} & \mathbf{K}_{lu} \\ \mathbf{K}_{lu}^T & \mathbf{K}_{uu} \end{pmatrix}$ where $\mathbf{K}_{ll}$ is the kernel matrix for the labeled data, $\mathbf{K}_{lu}$ is for the labeled and unlabeled data and $\mathbf{K}_{uu}$ is for the unlabeled data. Then we can rewrite the objective function of problem (18) as

$$f = \frac{1}{2\lambda_1}\mathrm{tr}\Big(\Big[\tilde{\mathbf{\Theta}}^T\mathbf{K}_{ll}\tilde{\mathbf{\Theta}} + 2\tilde{\mathbf{\Theta}}^T\mathbf{K}_{lu}\mathbf{\Gamma} + \mathbf{\Gamma}^T\mathbf{K}_{uu}\mathbf{\Gamma}\Big]\mathbf{\Omega}\Big) + \frac{1}{2\lambda_2}\mathrm{tr}(\mathbf{\Gamma}\mathbf{\Omega}\mathbf{\Gamma}^T) - \sum_{i=1}^{l}\sum_{j=1}^{m}\alpha_i^j, \tag{19}$$

where $\tilde{\mathbf{\Theta}} = \mathbf{Y} \odot \mathbf{\Theta}$. Since problem (18) is an unconstrained optimization problem with respect to $\mathbf{\Gamma}$, we calculate the derivative of f with respect to $\mathbf{\Gamma}$ as:

$$\frac{\partial f}{\partial \mathbf{\Gamma}} = \frac{1}{\lambda_2}\mathbf{\Gamma}\mathbf{\Omega} + \frac{1}{\lambda_1}\mathbf{K}_{lu}^T\tilde{\mathbf{\Theta}}\mathbf{\Omega} + \frac{1}{\lambda_1}\mathbf{K}_{uu}\mathbf{\Gamma}\mathbf{\Omega}.$$

Setting the derivative to 0, we obtain

$$\mathbf{\Gamma} = -\Big(\frac{\lambda_1}{\lambda_2}\mathbf{I}_u + \mathbf{K}_{uu}\Big)^{-1}\mathbf{K}_{lu}^T\tilde{\mathbf{\Theta}}. \tag{20}$$

We plug Eq. (20) into Eq. (19) and simplify the objective function of problem (18) as

$$\begin{aligned} f &= \frac{1}{2\lambda_1}\mathrm{tr}\Big(\tilde{\mathbf{\Theta}}^T\tilde{\mathbf{K}}_{ll}\tilde{\mathbf{\Theta}}\mathbf{\Omega}\Big) - \sum_{i=1}^{l}\sum_{j=1}^{m}\alpha_i^j \\ &= \frac{1}{2\lambda_1}(\mathrm{vec}(\tilde{\mathbf{\Theta}}))^T\Big(\mathbf{\Omega}\otimes\tilde{\mathbf{K}}_{ll}\Big)\mathrm{vec}(\tilde{\mathbf{\Theta}}) - \sum_{i=1}^{l}\sum_{j=1}^{m}\alpha_i^j \\ &= \frac{1}{2\lambda_1}(\mathrm{vec}(\mathbf{\Theta}))^T\mathrm{diag}(\mathrm{vec}(\mathbf{Y}))\Big(\mathbf{\Omega}\otimes\tilde{\mathbf{K}}_{ll}\Big)\mathrm{diag}(\mathrm{vec}(\mathbf{Y}))\mathrm{vec}(\mathbf{\Theta}) - \sum_{i=1}^{l}\sum_{j=1}^{m}\alpha_i^j, \end{aligned}$$

where $\tilde{\mathbf{K}}_{ll} = \mathbf{K}_{ll} - \mathbf{K}_{lu}(\frac{\lambda_1}{\lambda_2}\mathbf{I}_u + \mathbf{K}_{uu})^{-1}\mathbf{K}_{lu}^T$. It is easy to show that $\tilde{\mathbf{K}}_{ll}$ is a positive semidefinite (PSD) matrix according to the property of the Schur complement [Boyd and

Vandenberghe 2004]. Then problem (18) can be reformulated as

$$\begin{aligned}\min_{\boldsymbol{\alpha}} \quad & \frac{1}{2}\boldsymbol{\alpha}^T\hat{\mathbf{K}}_{ML}\boldsymbol{\alpha} - \sum_{i=1}^{l}\sum_{j=1}^{m}\alpha_i^j \\ \text{s.t.} \quad & \sum_{i=1}^{l}\alpha_i^j y_i^j = 0, \forall j \\ & 0 \le \alpha_i^j \le \frac{1}{l}, \forall i, j, \end{aligned} \qquad (21)$$

where $\boldsymbol{\alpha} = \text{vec}(\boldsymbol{\Theta})$ and $\hat{\mathbf{K}}_{ML} = \left(\frac{1}{\lambda_1}\boldsymbol{\Omega} \otimes \tilde{\mathbf{K}}_{ll}\right) \odot (\mathbf{y}\mathbf{y}^T)$. It is easy to show that $\hat{\mathbf{K}}_{ML}$ is a PSD matrix and so problem (21) is convex. Problem (21) is very similar to problem (10) except for the different definitions of $\hat{\mathbf{K}}_{ML}$ and $\tilde{\mathbf{K}}_{ML}$. So we still use the SMO algorithm in the appendix to solve problem (21).

From the formulation of $\tilde{\mathbf{K}}_{ll}$, we can define a data-dependent kernel $k'(\cdot,\cdot)$ as

$$k'(\mathbf{z}_1, \mathbf{z}_2) = k(\mathbf{z}_1, \mathbf{z}_2) - (\mathbf{k}_1^u)^T(\frac{\lambda_1}{\lambda_2}\mathbf{I}_u + \mathbf{K}_{uu})^{-1}\mathbf{k}_2^u, \qquad (22)$$

where $\mathbf{k}_i^u = (k(\mathbf{z}_i, \mathbf{x}_{l+1}), \ldots, k(\mathbf{z}_i, \mathbf{x}_{l+u}))^T$ for $i = 1, 2$. Then we can define the data-dependent multi-label kernel as $k'_{ML}((\mathbf{x}_i, j), (\mathbf{x}_p, q)) = \frac{\Omega_{jq}}{\lambda_1}k'(\mathbf{x}_i, \mathbf{x}_p)$ which describes the similarity betweem $\mathbf{x}_i$ and $\mathbf{x}_p$ when they hold the jth and qth labels, respectively.

Optimizing w.r.t. $\boldsymbol{\Omega}$ when $\mathbf{W}$ and $\mathbf{b}$ are fixed. When $\mathbf{W}$ and $\mathbf{b}$ are fixed, problem (14) becomes

$$\begin{aligned}&\min_{\boldsymbol{\Omega}} \lambda_1\text{tr}(\mathbf{W}\boldsymbol{\Omega}^{-1}\mathbf{W}^T) + \lambda_2\text{tr}(\mathbf{F}\boldsymbol{\Omega}^{-1}\mathbf{F}^T) \\ &\text{s.t. } \boldsymbol{\Omega} \succ 0 \\ &\quad \text{tr}(\boldsymbol{\Omega}) = 1.\end{aligned} \qquad (23)$$

Then we have

$$\begin{aligned}\text{tr}(\boldsymbol{\Omega}^{-1}\mathbf{A}) &= \text{tr}(\boldsymbol{\Omega}^{-1}\mathbf{A})\text{tr}(\boldsymbol{\Omega}) \\ &= \text{tr}((\boldsymbol{\Omega}^{-\frac{1}{2}}\mathbf{A}^{\frac{1}{2}})(\mathbf{A}^{\frac{1}{2}}\boldsymbol{\Omega}^{-\frac{1}{2}}))\text{tr}(\boldsymbol{\Omega}^{\frac{1}{2}}\boldsymbol{\Omega}^{\frac{1}{2}}) \\ &\ge (\text{tr}(\boldsymbol{\Omega}^{-\frac{1}{2}}\mathbf{A}^{\frac{1}{2}}\boldsymbol{\Omega}^{\frac{1}{2}}))^2 = (\text{tr}(\mathbf{A}^{\frac{1}{2}}))^2,\end{aligned}$$

where $\mathbf{A} = \lambda_1\mathbf{W}^T\mathbf{W} + \lambda_2\mathbf{F}^T\mathbf{F}$. As before, the first equality holds because of the last constraint in problem (23) and the last inequality holds because of the Cauchy-Schwarz inequality for the Frobenius norm. Moreover, $\text{tr}(\boldsymbol{\Omega}^{-1}\mathbf{A})$ attains its minimum value $(\text{tr}(\mathbf{A}^{\frac{1}{2}}))^2$ if and only if $\boldsymbol{\Omega}^{-\frac{1}{2}}\mathbf{A}^{\frac{1}{2}} = a\boldsymbol{\Omega}^{\frac{1}{2}}$ for some constant a and $\text{tr}(\boldsymbol{\Omega}) = 1$. So we can get the analytical solution

$$\boldsymbol{\Omega} = \frac{\left(\lambda_1\mathbf{W}^T\mathbf{W} + \lambda_2\mathbf{F}^T\mathbf{F}\right)^{\frac{1}{2}}}{\text{tr}\left(\,(\lambda_1\mathbf{W}^T\mathbf{W} + \lambda_2\mathbf{F}^T\mathbf{F})^{\frac{1}{2}}\,\right)}.$$

From this solution, we can see how the unlabeled data can help to estimate $\boldsymbol{\Omega}$. In problem (21), we only calculate the optimal $\boldsymbol{\alpha}$ (or $\boldsymbol{\Theta}$) but not $\mathbf{W}$ and $\mathbf{F}$. We will show how to compute $\mathbf{W}$ and $\mathbf{F}$ from $\boldsymbol{\alpha}$. From Eq. (17), we can get $\mathbf{W} = \frac{1}{\lambda_1}\boldsymbol{\Omega}\Big[\phi(\mathbf{X}_l)\tilde{\boldsymbol{\Theta}} + \phi(\mathbf{X}_u)\boldsymbol{\Gamma}\Big]$

where $\phi(\mathbf{X}_l) = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_l)]$. By using Eq. (20), we can obtain

$$\mathbf{W} = \frac{1}{\lambda_1}\Big[\mathbf{X}_l - \mathbf{X}_u\Big(\frac{\lambda_1}{\lambda_2}\mathbf{I}_u + \mathbf{K}_{uu}\Big)^{-1}\mathbf{K}_{lu}^T\Big]\tilde{\mathbf{\Theta}}\mathbf{\Omega}.$$

Then we can calculate $\mathbf{W}^T\mathbf{W}$ as

$$\begin{aligned}&\mathbf{W}^T\mathbf{W}\\ =&\frac{1}{\lambda_1^2}\mathbf{\Omega}\tilde{\mathbf{\Theta}}^T\Big[\mathbf{K}_{ll} - \mathbf{K}_{lu}\big(\frac{\lambda_1}{\lambda_2}\mathbf{I}_u + \mathbf{K}_{uu}\big)^{-1}\mathbf{K}_{lu}^T - \frac{\lambda_1}{\lambda_2}\mathbf{K}_{lu}\big(\frac{\lambda_1}{\lambda_2}\mathbf{I}_u + \mathbf{K}_{uu}\big)^{-2}\mathbf{K}_{lu}^T\Big]\tilde{\mathbf{\Theta}}\mathbf{\Omega}.\end{aligned}$$

According to the first constraint of problem (14), we can calculate $\mathbf{F}$ as

$$\mathbf{F} = \phi(\mathbf{X}_u)^T\mathbf{W} = \frac{1}{\lambda_1}\Big[\mathbf{K}_{lu}^T - \mathbf{K}_{uu}\Big(\frac{\lambda_1}{\lambda_2}\mathbf{I}_u + \mathbf{K}_{uu}\Big)^{-1}\mathbf{K}_{lu}^T\Big]\tilde{\mathbf{\Theta}}\mathbf{\Omega}.$$

5. EXPERIMENTS

In this section, we compare MLRL and SSMLRL with some state-of-the-art multi-label learning methods on several real-world applications.

5.1 Performance Measures

Since each data point has multiple labels, performance evaluation in multi-label learning is much more complicated than that in traditional single-label learning. In this paper, five performance measures designed for multi-label learning from [Schapire and Singer 2000; Tsoumakas and Katakis 2007] are used, i.e. *Hamming loss*, *one-error*, *coverage*, *ranking loss* and *average precision*.

Given a multi-label dataset $\mathcal{D} = \{(\mathbf{x}_i, Y_i)\}_{i=1}^n$ where Y_i is the set of labels associated with $\mathbf{x}_i$, the five measures are defined as follows. Here we assume that $h(\mathbf{x}_i)$ returns the predictions of $\mathbf{x}_i$, $h(\mathbf{x}_i, y)$ returns the confidence that y is a label associated with $\mathbf{x}_i$, and $rank^h(\mathbf{x}_i, y)$ returns the rank of y derived from $h(\mathbf{x}_i, y)$.

(a) *Hamming loss:*

$$\text{hloss}(h) = \frac{1}{mn}\sum_{i=1}^n |h(\mathbf{x}_i) \bigtriangleup Y_i| \qquad (24)$$

Here $\bigtriangleup$ denotes the symmetric difference between two sets. The Hamming loss measures the number of times that an instance-label pair is misclassified.

(b) *One-error:*

$$\text{one-error}(h) = \frac{1}{n}\sum_{i=1}^n I\big([\arg\max_y h(\mathbf{x}_i, y)] \notin Y_i\big) \qquad (25)$$

Here for predicate π, $I(\pi)$ equals 1 if π holds and 0 otherwise. The one-error measures the number of times in which the top-ranked label is not in the set of proper labels of the data point.

(c) *Coverage:*

$$\text{coverage}(h) = \frac{1}{n}\sum_{i=1}^n \max_{y\in Y_i} rank^h(\mathbf{x}_i, y) - 1 \qquad (26)$$

The coverage measures the number of steps needed, on average, to move down the label list in order to cover all the proper labels of the data point.

(d) *Ranking loss:*

$$\text{rloss}(h) = \frac{1}{n}\sum_{i=1}^{n} \frac{|R_i|}{|Y_i||\bar{Y}_i|} \tag{27}$$

Here $R_i = \{(y_1, y_2) \mid h(\mathbf{x}_i, y_1) \leq h(\mathbf{x}_i, y_2), (y_1, y_2) \in Y_i \times \bar{Y}_i\}$ and $\bar{Y}_i$ denotes the complementary set of Y_i. The ranking loss measures the average fraction of label pairs that are misordered for the data point.

(e) *Average precision:*

$$\text{avgprec}(h) = \frac{1}{n}\sum_{i=1}^{n} \frac{1}{|Y_i|} \sum_{y \in Y_i} \frac{|P_i(y)|}{rank^h(\mathbf{x}_i, y)} \tag{28}$$

Here $P_i(y) = \{y' \mid rank^h(\mathbf{x}_i, y') \leq rank^h(\mathbf{x}_i, y), y' \in Y_i\}$. The average precision measures the average fraction of proper labels ranked above a particular label $y \in Y_i$.

For the first four performance measures, the smaller the value the better the performance. For the average precision, on the other hand, a larger value implies a better performance.

5.2 Data Sets

Nine multi-label data sets are used in our experiments. Their characteristics are briefly summarized in Table I. Given a multi-label data set $\mathcal{D}$, we use $|\mathcal{D}|$, $dim(\mathcal{D})$ and $L(\mathcal{D})$ to denote its number of data points, data dimensionality and number of all distinct labels. Moreover, the following multi-label statistics from [Read et al. 2009] are also shown in Table I:

—Label cardinality $LCard(\mathcal{D}) = \frac{1}{n}\sum_{i=1}^{n} |Y_i|$, which measures the average number of labels in $\mathcal{D}$;

—Label density $LDen(\mathcal{D}) = \frac{LCard(\mathcal{D})}{L(\mathcal{D})}$, which normalizes $LCard(\mathcal{D})$ by the number of all distinct labels;

—Number of distinct label sets $DLS(\mathcal{D}) = |\{Y|(\mathbf{x}, Y) \in \mathcal{D}\}|$, which represents the number of distinct label combinations in $\mathcal{D}$;

—Proportion of distinct label sets $PDLS(\mathcal{D}) = \frac{DLS(\mathcal{D})}{|D|}$, which normalizes $DLS(\mathcal{D})$ by the number of data points in D.

As shown in Table I, four medium-scale data sets (emotions, image, scene, yeast) as well as five large-scale data sets (from rcv1(subset 1) to rcv1(subset 5)) are used in our experiments. Moreover, dimensionality reduction is performed on rcv1(subset 1) to rcv1(subset 5), where we retain the top 2% of features with highest document frequency as [Zhang and Zhang 2010] did.

For the emotions, image, scene, yeast data sets, the kernel we used is the RBF kernel and the linear kernel is adopted for rcv1 data sets. The width parameter used in the RBF kernel is set to be the mean of pairwise Euclidean distances of the whole training data, which include labeled data in supervised setting and labeled and unlabeled data in semi-supervised setting. The candidate set for the regularization parameter λ in MLRL is $\{0.01, 0.1, 1, 10, 100\}$ and the same candidate set is used for λ_1 and λ_2 in SSMLRL.

Table I. Characteristics of the data sets used in the experiments

Data Set	$\lvert\mathcal{D}\rvert$	$dim(\mathcal{D})$	$L(\mathcal{D})$	$LCard(\mathcal{D})$	$LDen(\mathcal{D})$	$DLS(\mathcal{D})$	$PDLS(\mathcal{D})$
emotions	593	72	6	1.869	0.311	27	0.046
image	2000	294	5	1.236	0.247	20	0.010
scene	2407	294	6	1.074	0.179	15	0.006
yeast	2417	103	14	4.237	0.303	198	0.082
rcv1(subset 1)	6000	944	101	2.880	0.029	1028	0.171
rcv1(subset 2)	6000	944	101	2.634	0.026	954	0.159
rcv1(subset 3)	6000	944	101	2.614	0.026	939	0.157
rcv1(subset 4)	6000	944	101	2.484	0.025	816	0.136
rcv1(subset 5)	6000	944	101	2.642	0.026	946	0.158

5.3 Experimental Results for the Supervised Setting

In this section, we evaluate our method under the supervised setting. The methods compared are BP-ML [Zhang and Zhou 2006] which is a multi-label neural network, ML-KNN [Zhang and Zhou 2007] which is a lazy learner for multi-label learning, two maximum margin classifiers, namely, BSVM [Boutell et al. 2004] and RankSVM [Elisseeff and Weston 2001], LEAD [Zhang and Zhang 2010] which first uses a Bayesian network to learn the label dependency and then learns multiple binary classifiers based on the structure of the Bayesian network. For BSVM, LibSVM[4] [Chang and Lin 2001] is used. Moreover, the hyper-parameters suggested in the respective references in the literature are used for the compared methods: For BSVM and RankSVM, the model parameters such as the regularization parameters are learned using the cross-training strategy; for ML-KNN, the number of nearest neighbors is set to 10 and Euclidean distance is used as the distance metric; For BP-ML, the number of hidden units is set to 20% of the dimensionality and the number of training epochs is set to 100.

Ten-fold cross-validation is performed on each data set to evaluate the performance of all compared methods. The results are reported from Tables II to VI. Pairwise t-tests at 5% significance level are conducted between all the methods to statistically measure the significance of performance difference. Whenever our method MLRL achieves significantly better/worse performance than the compared method on any data set, a win/loss is counted and a marker •/○ is shown in the corresponding table; otherwise the situation is treated as a tie and no marker is shown. The resulting win/tie/loss counts for MLRL against the compared methods are summarized in Table VII.

As shown in Table VII, MLRL is significantly superior to the compared methods in most cases: 86.7% (BSVM), 80.0% (RankSVM), 75.6% (ML-KNN), 86.7% (BP-ML) and 40.0% (LEAD), and is inferior to them in much fewer cases: 0.0% (BSVM), 2.2% (RankSVM), 0.0% (ML-KNN), 2.2% (BP-ML) and 2.2% (LEAD). These results show that MLRL is competitive with respect to the state-of-the-art methods.

5.3.1 *Analysis of Learned Label Correlation Matrices.*

For the emotions data, the mean label correlation matrix over 10 trials, calculated from the label covariance matrices over 10 trials, is reported in Table VIII. We can find that the correlations between 'happy-pleased' and all other emotions are near 0, meaning that 'happy-pleased' is uncorrelated with other emotions. This is easy to understand because other emotions are not

[4] http://www.csie.ntu.edu.tw/∼cjlin/libsvmtools/multilabel/

Table II. Performance of each algorithm in terms of *Hamming loss* under the *supervised setting*. For each data set, the upper row shows the mean over 10 trials for each method and the lower row shows the corresponding standard deviation. ●/○ indicates whether MLRL is statistically superior/inferior to the compared algorithm by pairwise t-test at 5% significance level.

Data Set	MLRL	BSVM	RankSVM	ML-KNN	BP-ML	LEAD
emotions	0.1880	0.2364●	0.2014●	0.2162●	0.2317●	0.2085●
	0.0139	0.0407	0.0289	0.0235	0.0175	0.0215
image	0.1739	0.2835●	0.2100●	0.2476●	0.1733	0.1810
	0.0102	0.0133	0.0354	0.0204	0.0077	0.0155
scene	0.1039	0.1596●	0.1337●	0.1880●	0.0889○	0.1260
	0.0104	0.0099	0.0106	0.0262	0.0118	0.0132
yeast	0.1986	0.2423●	0.2029	0.2132●	0.2143●	0.2102●
	0.0048	0.0310	0.0053	0.0053	0.0034	0.0038
rcv1(subset 1)	0.0265	0.0280	0.0295●	0.0301●	0.0332●	0.0278
	0.0013	0.0025	0.0038	0.0024	0.0023	0.0017
rcv1(subset 2)	0.0210	0.0291●	0.0315●	0.0247	0.0312●	0.0246●
	0.0011	0.0019	0.0021	0.0017	0.0018	0.0012
rcv1(subset 3)	0.0229	0.0287●	0.0273●	0.0257	0.0341●	0.0256
	0.0015	0.0021	0.0017	0.0020	0.0029	0.0011
rcv1(subset 4)	0.0208	0.0287●	0.0308●	0.0280●	0.0321●	0.0231
	0.0017	0.0016	0.0024	0.0021	0.0023	0.0015
rcv1(subset 5)	0.0231	0.0297●	0.0281●	0.0273●	0.0312●	0.0237
	0.0019	0.0021	0.0020	0.0018	0.0019	0.0016

Table III. Performance of each algorithm in terms of *one-error* under the *supervised setting*. For each data set, the upper row shows the mean over 10 trials for each method and the lower row shows the corresponding standard deviation. ●/○ indicates whether MLRL is statistically superior/inferior to the compared algorithm by pairwise t-test at 5% significance level.

Data Set	MLRL	BSVM	RankSVM	ML-KNN	BP-ML	LEAD
emotions	0.2548	0.4128●	0.3924●	0.3145	0.3913●	0.3011
	0.0644	0.0360	0.0727	0.0781	0.0522	0.0631
image	0.3080	0.3565●	0.3435●	0.3370	0.3750●	0.3095
	0.0265	0.0129	0.0973	0.0619	0.0248	0.0223
scene	0.2472	0.3158●	0.2922●	0.2510	0.6289●	0.2502
	0.0375	0.0062	0.0428	0.0275	0.0360	0.0476
yeast	0.2267	0.2944●	0.2230	0.2292	0.2396	0.2590●
	0.0164	0.0289	0.0260	0.0308	0.0191	0.0259
rcv1(subset 1)	0.4120	0.4413●	0.4632●	0.6123●	0.8139●	0.4416●
	0.0153	0.0147	0.0182	0.0187	0.0213	0.0164
rcv1(subset 2)	0.3937	0.4421●	0.4210	0.5817●	0.7365●	0.4212
	0.0159	0.0186	0.0179	0.0186	0.0251	0.0171
rcv1(subset 3)	0.3985	0.4315●	0.4277●	0.5366●	0.6789●	0.3995
	0.0161	0.0191	0.0153	0.0153	0.0201	0.0147
rcv1(subset 4)	0.3356	0.4151●	0.4310●	0.5001●	0.6912●	0.3923●
	0.0143	0.0173	0.0168	0.0191	0.0204	0.0162
rcv1(subset 5)	0.4081	0.4812●	0.4537●	0.5458●	0.6819●	0.4317
	0.0178	0.0169	0.0185	0.0194	0.0228	0.0146

about ‘happiness’. The mean label correlation matrices for the image and scene datasets are shown in Tables IX and X, respectively. For the image data, ‘desert’ has negative correlations with ‘sea’ and ‘tree’ because we seldom see water and trees in a desert and we seldom see a desert when we see water or trees. For the scene data, ‘beach’ has nearly zero

Table IV. Performance of each algorithm in terms of *coverage* under the *supervised setting*. For each data set, the upper row shows the mean over 10 trials for each method and the lower row shows the corresponding standard deviation. ●/○ indicates whether MLRL is statistically superior/inferior to the compared algorithm by pairwise t-test at 5% significance level.

Data Set	MLRL	BSVM	RankSVM	ML-KNN	BP-ML	LEAD
emotions	1.7686	2.0110●	1.9814●	1.9131●	2.0336●	1.9711●
	0.1148	0.1592	0.1942	0.1517	0.1806	0.1623
image	0.9365	1.1340●	1.0535●	1.8385●	1.0150●	0.9928
	0.0526	0.0656	0.0468	0.0570	0.0532	0.0622
scene	0.5206	0.5184	0.4974○	0.5481	0.8202●	0.5223
	0.0849	0.0730	0.0882	0.1392	0.0807	0.0829
yeast	5.4623	6.5355●	6.3599●	6.3604●	6.3247●	5.8204●
	0.2045	0.2040	0.0938	0.1108	0.1692	0.2749
rcv1(subset 1)	12.0141	21.5187●	22.1391●	20.0142●	28.1726●	14.2749●
	0.6810	0.8157	0.9105	1.0310	0.8591	0.7230
rcv1(subset 2)	11.0141	22.2209●	23.4208●	21.4108●	29.2992●	12.9247●
	0.7821	0.8297	0.9201	0.9184	0.9104	0.8192
rcv1(subset 3)	10.2090	22.0183●	22.1940●	22.1483●	25.4102●	11.8109
	0.8109	0.9109	0.8104	0.9174	0.8197	0.9179
rcv1(subset 4)	12.8420	20.0841●	21.8420●	22.8042●	21.4280●	10.2084○
	0.7013	0.8023	0.9018	0.9723	0.9731	0.9739
rcv1(subset 5)	11.4297	22.2389●	22.2979●	23.4203●	26.0130●	12.4200
	0.7912	0.8791	0.8198	0.8941	0.8012	0.8929

Table V. Performance of each algorithm in terms of *ranking loss* under the *supervised setting*. For each data set, the upper row shows the mean over 10 trials for each method and the lower row shows the corresponding standard deviation. ●/○ indicates whether MLRL is statistically superior/inferior to the compared algorithm by pairwise t-test at 5% significance level.

Data Set	MLRL	BSVM	RankSVM	ML-KNN	BP-ML	LEAD
emotions	0.1574	0.1843	0.2011●	0.1779	0.1843	0.1620
	0.0371	0.0516	0.0554	0.0227	0.0324	0.0204
image	0.1661	0.1890●	0.1960●	0.1903●	0.1854●	0.1712
	0.0137	0.0139	0.0614	0.0419	0.0145	0.0103
scene	0.0862	0.0965	0.0826	0.1029●	0.2863●	0.0929
	0.0146	0.0147	0.0156	0.0280	0.0144	0.0120
yeast	0.1615	0.2575●	0.1677	0.1724	0.1716	0.1719
	0.0146	0.0376	0.0070	0.0083	0.0094	0.0102
rcv1(subset 1)	0.0510	0.0911●	0.1020●	0.1102●	0.1448●	0.0722●
	0.0031	0.0042	0.0037	0.0042	0.0030	0.0041
rcv1(subset 2)	0.0482	0.0962●	0.1043●	0.1009●	0.1501●	0.0634●
	0.0030	0.0041	0.0031	0.0047	0.0045	0.0043
rcv1(subset 3)	0.0448	0.0802●	0.1058●	0.1001●	0.1502●	0.0610●
	0.0032	0.0046	0.0042	0.0045	0.0038	0.0047
rcv1(subset 4)	0.0402	0.0940●	0.1029●	0.1004●	0.1442●	0.0589●
	0.0041	0.0044	0.0035	0.0042	0.0047	0.0046
rcv1(subset 5)	0.0552	0.0892●	0.1001●	0.1021●	0.1542●	0.0521○
	0.0040	0.0041	0.0031	0.0046	0.0038	0.0045

correlation with ‘sunset’ because these two scenes seldom appear together in an image. We can see that the knowledge inferred from the label correlation matrices matches people’s intuition and hence MLRL has higher interpretability.

Table VI. Performance of each algorithm in terms of *average precision* under the *supervised setting*. For each data set, the upper row shows the mean over 10 trials for each method and the lower row shows the corresponding standard deviation. ●/○ indicates whether MLRL is statistically superior/inferior to the compared algorithm by pairwise t-test at 5% significance level.

Data Set	MLRL	BSVM	RankSVM	ML-KNN	BP-ML	LEAD
emotions	0.8079	0.7873	0.6683●	0.7707	0.7149●	0.7920
	0.0420	0.0332	0.0491	0.0300	0.0282	0.0358
image	0.7988	0.7365●	0.7639	0.5925●	0.7840	0.7610●
	0.0154	0.0060	0.0587	0.0358	0.0149	0.0203
scene	0.8506	0.8449	0.8471	0.8391	0.8294●	0.8492
	0.0221	0.0063	0.0250	0.0209	0.0224	0.0290
yeast	0.7825	0.7459●	0.7620	0.7539●	0.7590●	0.7684
	0.0121	0.0151	0.0144	0.0168	0.0139	0.0148
rcv1(subset 1)	0.6202	0.5209●	0.5710●	0.4804●	0.3820●	0.5880●
	0.0084	0.0093	0.0090	0.0130	0.0128	0.0092
rcv1(subset 2)	0.6431	0.5402●	0.5484●	0.4841●	0.3274●	0.6340●
	0.0081	0.0098	0.0093	0.0125	0.0121	0.0085
rcv1(subset 3)	0.6342	0.5810●	0.5452●	0.5101●	0.4182●	0.6124
	0.0124	0.0140	0.0185	0.0181	0.0103	0.0129
rcv1(subset 4)	0.6642	0.6010●	0.5602●	0.5901●	0.4353●	0.6552
	0.0108	0.0104	0.0180	0.0187	0.0142	0.0135
rcv1(subset 5)	0.6242	0.5929●	0.5621●	0.5521●	0.3891●	0.6104●
	0.0131	0.0131	0.0171	0.0162	0.0124	0.0146

Table VII. The win/tie/loss results for MLRL against the compared methods in terms of different performance measures under the *supervised setting*.

Performance Measure	BSVM	RankSVM	ML-KNN	BP-ML	LEAD
Hamming loss	8/1/0	8/1/0	7/2/0	7/1/1	3/6/0
one-error	9/0/0	7/2/0	5/4/0	8/1/0	3/6/0
coverage	8/1/0	8/0/1	8/1/0	9/0/0	4/5/0
ranking loss	7/2/0	7/2/0	7/2/0	7/2/0	4/4/1
average precision	7/2/0	6/3/0	7/2/0	8/1/0	4/5/0
Total	39/6/0	36/8/1	34/11/0	39/5/1	18/26/1

Table VIII. Mean label correlation matrix learned from the music emotions data. The labels are: L1: amazed-surprised; L2: happy-pleased; L3: relaxing-calm; L4: quiet-still; L5: sad-lonely; L6: angry-aggressive.

	L1	L2	L3	L4	L5	L6
L1	1.0000	0.0000	-0.7300	-0.3389	-0.2858	0.1725
L2	0.0000	1.0000	0.0000	-0.0000	-0.0000	-0.0000
L3	-0.7300	0.0000	1.0000	0.2674	0.1638	-0.7178
L4	-0.3389	-0.0000	0.2674	1.0000	0.8687	-0.2799
L5	-0.2858	-0.0000	0.1638	0.8687	1.0000	-0.1181
L6	0.1725	-0.0000	-0.7178	-0.2799	-0.1181	1.0000

5.4 Experimental Results for the Semi-Supervised Setting

In this section, we evaluate our method under the semi-supervised setting. The methods compared are MLGRF [Chen et al. 2008; Zha et al. 2009] which generalizes the conventional Gaussian random field [Zhu et al. 2003] to the multi-label setting, MLLGC [Chen et al. 2008; Zha et al. 2009] which generalizes the conventional local and global consisten-

Table IX. Mean label correlation matrix learned from the image data.

	desert	mountain	sea	sunset	tree
desert	1.0000	-0.2417	-0.2561	-0.3104	-0.2759
mountain	-0.2417	1.0000	-0.3600	-0.1351	-0.1184
sea	-0.2561	-0.3600	1.0000	0.0295	-0.3230
sunset	-0.3104	-0.1351	0.0295	1.0000	-0.2414
tree	-0.2759	-0.1184	-0.3230	-0.2414	1.0000

Table X. Mean label correlation matrix learned from the scene data.

	beach	sunset	foliage	field	hill	urban
beach	1.0000	-0.0208	-0.1709	-0.2714	-0.2036	-0.2669
sunset	-0.0208	1.0000	-0.4559	-0.0270	-0.2017	-0.2331
foliage	-0.1709	-0.4559	1.0000	-0.2124	-0.0369	-0.0774
field	-0.2714	-0.0270	-0.2124	1.0000	-0.0841	-0.1421
hill	-0.2036	-0.2017	-0.0369	-0.0841	1.0000	-0.4337
urban	-0.2669	-0.2331	-0.0774	-0.1421	-0.4337	1.0000

cy method [Zhou et al. 2003] to the multi-label setting, and CNMF [Liu et al. 2006] which is formulated as a constrained nonnegative matrix factorization method. Moreover, MLRL is used as a benchmark method in this comparison.

Since all compared methods including MLGRF, MLLGC and CNMF are transductive methods, we conduct experiments under the transductive setting even though SSMLRL is in fact an inductive method. We randomly choose 20% of each data set as the labeled data and the rest as unlabeled data. We make random splits for 10 trials and report the mean and standard deviation for each performance measure over the 10 trials. The results are reported in Tables XI to XV with respect to different performance measures. Similar to the experimental settings in supervised learning, pairwise t-tests at 5% significance level are conducted between all the methods and a marker •/○ is shown in these tables to show whether SSMLRL is significantly better/worse than the compared methods. The win/tie/loss counts for MLRL against the compared methods on all data sets are summarized in Tables XVI.

As shown in Table XVI, SSMLRL is again significantly superior to the compared methods in most cases: 91.1% (MLRL), 48.9% (MLGRF), 73.3% (MLLGC) and 95.6% (CNMF), and is inferior to them in much few cases: 0.0% (MLRL), 13.3% (MLGRF), 11.1% (MLLGC) and 0.0% (CNMF). These results show that SSMLRL is competitive to the state-of-the-art methods for the semi-supervised multi-label learning.

Moreover, to our surprise, Tables XI to XV show that MLRL, even as a supervised method without exploiting unlabeled data, outperforms the semi-supervised learning methods in the literature.

5.5 Computational Details

We plot the change in values of the objective functions in problem (6) and (14) in Figure 1(a) and 1(b) on the emotion data. We find that the objective function value decreases rapidly and then levels off under the two settings, showing the fast convergence of the algorithm which takes no more than 15 iterations.

Moreover, we record the mean of the running time for solving problem (6) and (14)

Table XI. Performance of each algorithm in terms of *Hamming loss* under the *semi-supervised setting*. For each data set, the upper row shows the mean over 10 trials for each method and the lower row shows the corresponding standard deviation. ●/○ indicates whether SSMLRL is statistically superior/inferior to the compared algorithm by pairwise t-test at 5% significance level.

Data Set	SSMLRL	MLRL	MLGRF	MLLGC	CNMF
emotions	0.2449	0.2671	0.2989●	0.3860●	0.3153●
	0.0110	0.0170	0.0097	0.0225	0.0031
image	0.1787	0.2345●	0.2008●	0.3417●	0.2468●
	0.0061	0.0070	0.0063	0.0197	0.0012
scene	0.0983	0.1815●	0.1129●	0.1171●	0.1791●
	0.0035	0.0118	0.0077	0.0065	0.0004
yeast	0.2098	0.2578●	0.2171	0.2557●	0.3026●
	0.0027	0.0101	0.0068	0.0133	0.0013
rcv1(subset 1)	0.0385	0.0596	0.2673●	0.0433	0.1356●
	0.0066	0.0169	0.0735	0.0095	0.0087
rcv1(subset 2)	0.0361	0.0565●	0.2376●	0.0662●	0.1069●
	0.0021	0.0123	0.0782	0.0104	0.0064
rcv1(subset 3)	0.0302	0.0526●	0.2023●	0.0520	0.1143●
	0.0032	0.0147	0.0812	0.0134	0.0031
rcv1(subset 4)	0.0320	0.0580●	0.2094●	0.0533●	0.1223●
	0.0034	0.0131	0.0729	0.0106	0.0079
rcv1(subset 5)	0.0349	0.0543●	0.2480●	0.0724●	0.1353●
	0.0032	0.0142	0.0808	0.0142	0.0098

Table XII. Performance of each algorithm in terms of *one-error* under the *semi-supervised setting*. For each data set, the upper row shows the mean over 10 trials for each method and the lower row shows the corresponding standard deviation. ●/○ indicates whether SSMLRL is statistically superior/inferior to the compared algorithm by pairwise t-test at 5% significance level.

Data Set	SSMLRL	MLRL	MLGRF	MLLGC	CNMF
emotions	0.3580	0.4008●	0.4559●	0.4660●	0.5800●
	0.0286	0.0247	0.0261	0.0155	0.0142
image	0.3307	0.5969●	0.3636●	0.3809●	0.4491●
	0.0178	0.0688	0.0112	0.0491	0.0107
scene	0.2526	0.5415●	0.2629	0.2843●	0.3367●
	0.0097	0.0655	0.0108	0.0064	0.0388
yeast	0.2435	0.2616●	0.2540	0.2504	0.9006●
	0.0074	0.0057	0.0052	0.0063	0.0314
rcv1(subset 1)	0.4700	0.4726	0.4724	0.5054●	0.7748●
	0.0055	0.0064	0.0046	0.0059	0.0038
rcv1(subset 2)	0.4276	0.4495●	0.4283	0.5071●	0.7602●
	0.0102	0.0117	0.0066	0.0065	0.0137
rcv1(subset 3)	0.4601	0.4942●	0.4818●	0.5150●	0.7620●
	0.0060	0.0081	0.0039	0.0053	0.0080
rcv1(subset 4)	0.4528	0.4958●	0.4682	0.4902●	0.7773●
	0.0108	0.0101	0.0080	0.0080	0.0182
rcv1(subset 5)	0.4642	0.5104●	0.4786	0.5108●	0.7802●
	0.0083	0.0069	0.0068	0.0059	0.0065

after 100 trials for each dataset in Table XVII. The platform to run the experiments is Intel i7 CPU 2.7GHz with 8GB RAM and we use Matlab 2009b for implementation and experiments. From Table XVII, we can see that our proposed optimization method is very efficient on all the datasets used.

Table XIII. Performance of each algorithm in terms of *coverage* under the *semi-supervised setting*. For each data set, the upper row shows the mean over 10 trials for each method and the lower row shows the corresponding standard deviation. ●/○ indicates whether SSMLRL is statistically superior/inferior to the compared algorithm by pairwise t-test at 5% significance level.

Data Set	SSMLRL	MLRL	MLGRF	MLLGC	CNMF
emotions	2.1116	2.3295●	2.4749●	2.4352●	2.7709●
	0.0772	0.1643	0.0785	0.0839	0.0767
image	1.0007	1.6538●	1.0702	1.1013	1.2196●
	0.0517	0.2100	0.0309	0.0300	0.0351
scene	0.5679	1.3858●	0.5329	0.5624	0.6392●
	0.0234	0.2681	0.0217	0.0117	0.0507
yeast	6.8385	7.3589●	6.5325○	6.3800○	11.7836
	0.0673	0.1677	0.0477	0.0519	0.0759
rcv1(subset 1)	20.1861	30.0619●	42.2791●	23.4750●	31.6373●
	0.4766	2.0452	2.0387	0.4616	0.3448
rcv1(subset 2)	19.2883	27.3986●	39.2849●	21.9597●	30.2385●
	0.5408	1.7177	1.3993	0.4678	0.1519
rcv1(subset 3)	20.8042	27.5023●	40.4029●	20.5803	29.8201●
	0.5058	2.0183	1.9804	0.8042	0.5039
rcv1(subset 4)	20.5031	28.5210●	37.0653●	18.4209○	28.0028●
	0.6004	1.8903	1.5032	0.5087	0.8650
rcv1(subset 5)	21.0287	30.0298●	41.5820●	24.5032●	28.2058●
	0.5801	1.8209	1.8294	0.4616	0.5039

Table XIV. Performance of each algorithm in terms of *ranking loss* under the *semi-supervised setting*. For each data set, the upper row shows the mean over 10 trials for each method and the lower row shows the corresponding standard deviation. ●/○ indicates whether SSMLRL is statistically superior/inferior to the compared algorithm by pairwise t-test at 5% significance level.

Data Set	SSMLRL	MLRL	MLGRF	MLLGC	CNMF
emotions	0.2285	0.2371	0.3138●	0.3101●	0.4019●
	0.0152	0.0179	0.0171	0.0158	0.0211
image	0.1834	0.3136●	0.2003	0.2095●	0.2351●
	0.0116	0.0556	0.0069	0.0070	0.0080
scene	0.0950	0.1511●	0.0894○	0.0955	0.1091
	0.0044	0.0983	0.0041	0.0021	0.0099
yeast	0.1937	0.2215●	0.1885○	0.1859○	0.7090●
	0.0025	0.0080	0.0015	0.0024	0.0103
rcv1(subset 1)	0.0960	0.1215●	0.0962	0.1555●	0.1647●
	0.0024	0.0049	0.0032	0.0027	0.0018
rcv1(subset 2)	0.0985	0.1201●	0.0953	0.1525●	0.1613●
	0.0031	0.0052	0.0030	0.0062	0.0020
rcv1(subset 3)	0.0853	0.1183●	0.0719○	0.1431●	0.1770●
	0.0024	0.0037	0.0021	0.0031	0.0051
rcv1(subset 4)	0.0949	0.1105●	0.0937	0.1363●	0.1561●
	0.0036	0.0048	0.0035	0.0060	0.0047
rcv1(subset 5)	0.0928	0.1081●	0.1052●	0.1461●	0.1587●
	0.0039	0.0051	0.0046	0.0046	0.0068

6. CONCLUSION

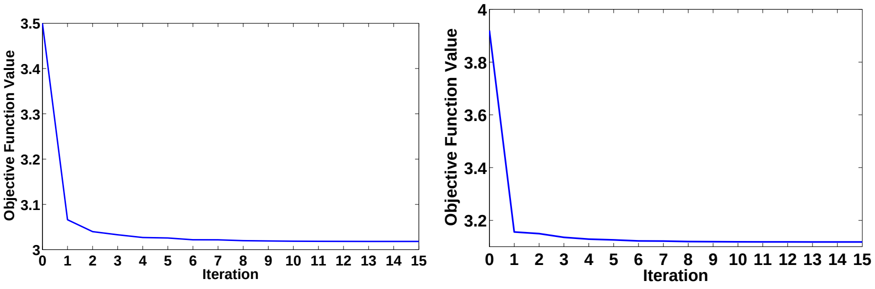
In this paper, we have proposed a new maximum margin method which learns the relationships between labels from data and utilizes them for multi-label learning. The optimiza-

Table XV. Performance of each algorithm in terms of *average precision* under the *semi-supervised setting*. For each data set, the upper row shows the mean over 10 trials for each method and the lower row shows the corresponding standard deviation. ●/○ indicates whether SSMLRL is statistically superior/inferior to the compared algorithm by pairwise t-test at 5% significance level.

Data Set	SSMLRL	MLRL	MLGRF	MLLGC	CNMF
emotions	0.7414	0.7029●	0.6689●	0.6694●	0.5992●
	0.0160	0.0521	0.0148	0.0112	0.0097
image	0.7837	0.6001●	0.7626	0.7522●	0.7118●
	0.0111	0.0571	0.0065	0.0080	0.0075
scene	0.8450	0.5638●	0.8436	0.8318●	0.8025●
	0.0051	0.0585	0.0063	0.0032	0.0205
yeast	0.7435	0.7224●	0.7393●	0.7391●	0.3035●
	0.0031	0.0076	0.0026	0.0030	0.0085
rcv1(subset 1)	0.4741	0.2676●	0.5306○	0.5134○	0.2525●
	0.0891	0.1481	0.0047	0.0040	0.0087
rcv1(subset 2)	0.4576	0.2524●	0.5704○	0.5534○	0.2903●
	0.0359	0.0855	0.0037	0.0036	0.0174
rcv1(subset 3)	0.5485	0.3402●	0.5369	0.5163●	0.2801●
	0.0858	0.0895	0.0085	0.0062	0.0071
rcv1(subset 4)	0.5808	0.3287●	0.5663	0.5603●	0.3052●
	0.0353	0.0780	0.0059	0.0044	0.0091
rcv1(subset 5)	0.5390	0.2959●	0.5251●	0.5158●	0.2803●
	0.0768	0.0874	0.0060	0.0042	0.0089

Table XVI. The win/tie/loss results for SSMLRL against the compared methods in terms of different performance measures under the *semi-supervised setting*.

Performance Measure	MLRL	MLGRF	MLLGC	CNMF
Hamming loss	7/2/0	8/1/0	7/2/0	9/0/0
one-error	8/1/0	3/6/0	8/1/0	9/0/0
coverage	9/0/0	6/2/1	4/3/2	8/1/0
ranking loss	8/1/0	2/4/3	7/1/1	8/1/0
average precision	9/0/0	3/4/2	7/0/2	9/0/0
Total	41/4/0	22/17/6	33/7/5	43/2/0



(a) MLRL (b) SSMLRL

Fig. 1. Convergence of objective function value for the music emotions data under the supervised and semi-supervised settings.

Table XVII. The mean of the running time (in second) for MLRL and SSMLRL after 100 trials for each dataset.

Data Set	MLRL	SSMLRL
emotions	3.6981	7.3062
image	6.3663	18.6122
scene	15.5381	33.2390
yeast	30.0544	39.7401
rcv1(subset 1)	1148.1750	1270.3306
rcv1(subset 2)	1101.2013	1311.1205
rcv1(subset 3)	1192.8701	1281.1087
rcv1(subset 4)	1151.2564	1219.5361
rcv1(subset 5)	1169.1250	1243.3612

tion problem is convex and it can be solved efficiently using an alternating method. Not only does MLRL give better performance when compared with other multi-label learning methods, but it also has better interpretability because the relationships between labels are explicitly represented by the label covariance matrix. Moreover, we also present a semi-supervised extension of MLRL by exploiting the useful information in the unlabeled data.

Prior knowledge about relationships between labels may exist in some applications, and hierarchical multi-label learning methods [Cesa-Bianchi et al. 2006; Rousu et al. 2006; Vens et al. 2008; Hariharan et al. 2010] can utilize such prior knowledge. In our future research, we will pursue this extension for MLRL.

Acknowledgments

Dit-Yan Yeung has been supported by General Research Fund 621310 from the Research Grants Council of Hong Kong.

REFERENCES

ARGYRIOU, A., EVGENIOU, T., AND PONTIL, M. 2008. Convex multi-task feature learning. *Machine Learning 73,* 3, 243–272.

ARGYRIOU, A., MICCHELLI, C. A., PONTIL, M., AND YING, Y. 2008. A spectral regularization framework for multi-task structure learning. In *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Vancouver, British Columbia, Canada, 25–32.

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. AND VANDENBERGHE, L. 2004. *Convex Optimization*. Cambridge University Press, New York, NY.

BUCAK, S. S., MALLAPRAGADA, P. K., JIN, R., AND JAIN, A. K. 2009. Efficient multi-label ranking for multi-class learning: Application to object recognition. In *Proceedings of the Twelth IEEE International Conference on Computer Vision*. Kyoto, Japan.

CESA-BIANCHI, N., GENTILE, C., AND ZANIBONI, L. 2006. Hierarchical classification: Combining Bayes with SVM. In *Proceedings of the Twenty-Third International Conference on Machine Learning*. Pittsburgh, Pennsylvania, USA, 177–184.

CHANG, C.-C. AND LIN, C.-J. 2001. *LIBSVM: a library for support vector machines*. Software available at http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

CHAPELLE, O., ZIEN, A., AND SCHÖLKOPF, B., Eds. 2006. *Semi-Supervised Learning*. MIT Press, Boston.

CHEN, G., SONG, Y., WANG, F., AND ZHANG, C. 2008. Semi-supervised multi-label learning by solving a sylvester equation. In *Proceedings of the SIAM International Conference on Data Mining*. Atlanta, Georgia, USA, 410–419.

CLARE, A. AND KING, R. D. 2001. Knowledge discovery in multi-label phenotype data. In *Proceedings of the fifth European Conference on Principles of Data Mining and Knowledge Discovery*. Freiburg, Germany, 42–53.

DINUZZO, F. AND FUKUMIZU, K. 2011. Learning low-rank output kernels. In *Proceedings of the 3rd Asian Conference on Machine Learning*. Taoyuan, Taiwan, 181–196.

DINUZZO, F., ONG, C. S., GEHLER, P. V., AND PILLONETTO, G. 2011. Learning output kernels with block coordinate descent. In *Proceedings of the 28th International Conference on Machine Learning*. Bellevue, Washington, USA, 49–56.

ELISSEEFF, A. AND WESTON, J. 2001. A kernel method for multi-labelled classification. In *Advances in Neural Information Processing Systems 14*, T. G. Dietterich, S. Becker, and Z. Ghahramani, Eds. Vancouver, British Columbia, Canada, 681–687.

FAN, R.-E., CHEN, P.-H., AND LIN, C.-J. 2005. Working set selection using second order information for training support vector machines. *Journal of Machine Learing Research 6*, 1889–1918.

FÜRNKRANZ, J., HÜLLERMEIER, E., MENCÍA, E. L., AND BRINKER, K. 2008. Multilabel classification via calibrated label ranking. *Machine Learning 73,* 2, 133–153.

GUPTA, A. K. AND NAGAR, D. K. 2000. *Matrix Variate Distributions*. Chapman & Hall.

HARIHARAN, B., ZELNIK-MANOR, L., VISHWANATHAN, S. V. N., AND VARMA, M. 2010. Large scale max-margin multi-label classification with priors. In *Proceedings of the 27th International Conference on Machine Learning*. Haifa, Israel, 423–430.

JI, S., TANG, L., YU, S., AND YE, J. 2010. A shared-subspace learning framework for multi-label classification. *ACM Transactions on Knowledge Discovery from Data 4,* 2, 1–29.

JOACHIMS, T. 1998. Making large-scale svm learning practical. In *Advances in Kernel Methods- Support Vector Learning*, B. Scholkopf, C. J. C. Burges, and A. J. Smola, Eds. MIT Press, Cambridge.

KWOK, J. 1999. Moderating the outputs of support vector machine classifiers. *IEEE Transactions on Neural Networks 10,* 5, 1018–1031.

LIU, Y., JIN, R., AND YANG, L. 2006. Semi-supervised multi-label learning by constrained non-negative matrix factorization. In *Proceedings of the Twenty-First National Conference on Artificial Intelligence*. Boston, Massachusetts.

READ, J., PFAHRINGER, B., HOLMES, G., AND FRANK, E. 2009. Classifier chains for multi-label classification. In *Proceedings of European Conference on Machine Learning and Knowledge Discovery in Databases*. Bled, Slovenia, 254–269.

ROUSU, J., SAUNDERS, C., SZEDMÁK, S., AND SHAWE-TAYLOR, J. 2006. Kernel-based learning of hierarchical multilabel classification models. *Journal of Machine Learning Research 7*, 1601–1626.

SCHAPIRE, R. E. AND SINGER, Y. 2000. Boostexter: A boosting-based system for text categorization. *Machine Learning 39,* 2–3, 135–168.

SEBER, G. A. F. 2007. *A Matrix Handbook for Statisticians*. Wiley-Interscience.

SREBRO, N., RENNIE, J. D. M., AND JAAKKOLA, T. 2004. Maximum-margin matrix factorization. In *Advances in Neural Information Processing Systems 17*. Vancouver, British Columbia, Canada.

TSOUMAKAS, G. AND KATAKIS, I. 2007. Multi-label classification: An overview. *International Journal of Data Warehousing and Mining 3,* 3, 1–13.

TSOUMAKAS, G., KATAKIS, I., AND VLAHAVAS, I. 2010. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*, O. Maimon and L. Rokach, Eds. Springer, Berlin.

VENS, C., STRUYF, J., SCHIETGAT, L., DZEROSKI, S., AND BLOCKEEL, H. 2008. Decision trees for hierarchical multi-label classification. *Machine Learning 73,* 2, 185–214.

ZHA, Z.-J., MEI, T., WANG, J., WANG, Z., AND HUA, X.-S. 2009. Graph-based semi-supervised learning with multiple labels. *Journal of Visual Communication and Image Representation 20,* 2, 97–103.

ZHANG, M.-L. AND ZHANG, K. 2010. Multi-label learning by exploiting label dependency. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Washington, DC, USA, 999–1008.

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 Engeering 18,* 10, 1338–1351.

ZHANG, M.-L. AND ZHOU, Z.-H. 2007. ML-KNN: A lazy learning approach to multi-label learning. *Pattern Recognition 40,* 7, 2038–2048.

ZHANG, Y. AND YEUNG, D.-Y. 2010. A convex formulation for learning task relationships in multi-task learning. In *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence*. Catalina Island, California, 733–742.

ZHOU, D., BOUSQUET, O., LAL, T. N., WESTON, J., AND SCHÖLKOPF, B. 2003. Learning with local and global consistency. In *Advances in Neural Information Processing Systems 16*, S. Thrun, L. K. Saul, and B. Schölkopf, Eds. Vancouver, British Columbia, Canada.

ZHU, X., GHAHRAMANI, Z., AND LAFFERTY, J. D. 2003. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the Twentieth International Conference on Machine Learning*. Washington, DC, USA, 912–919.

Appendix A

In this section, we provide details of the SMO-style algorithm for solving the dual problem (10).

The Lagrangian of problem (10) is given by

$$L = \frac{1}{2}\boldsymbol{\alpha}^T\tilde{\mathbf{K}}_{ML}\boldsymbol{\alpha} - \sum_{i=1}^{m}\sum_{j=1}^{n}\alpha_j^i - \sum_{i=1}^{m}\gamma_i\sum_{j=1}^{n}\alpha_j^i y_j^i - \sum_{i=1}^{m}\sum_{j=1}^{n}\mu_j^i\alpha_j^i + \sum_{i=1}^{m}\sum_{j=1}^{n}\nu_j^i(\alpha_j^i - \frac{1}{n}).$$

Let $f_j^i = \boldsymbol{\alpha}^T(\mathbf{k}_j^i \odot \mathbf{y}) - y_j^i$ where $\mathbf{k}_j^i$ is a column of $\mathbf{K}_{ML}$. The KKT conditions for the dual problem are

$$\frac{\partial L}{\partial \alpha_j^i} = (f_j^i - \gamma_i)y_j^i - \mu_j^i + \nu_j^i = 0, \mu_j^i \geq 0, \mu_j^i\alpha_j^i = 0, \nu_j^i \geq 0, \nu_j^i(\alpha_j^i - \frac{1}{n}) = 0.$$

These conditions can be simplified into the following three cases:
Case 1 ($\alpha_j^i = 0$):

$$\nu_j^i = 0, \mu_j^i \geq 0 \Rightarrow (f_j^i - \gamma_i)y_j^i \geq 0$$

Case 2 ($0 < \alpha_j^i < \frac{1}{n}$):

$$\nu_j^i = 0, \mu_j^i = 0 \Rightarrow (f_j^i - \gamma_i)y_j^i = 0$$

Case 3 ($\alpha_j^i = \frac{1}{n}$):

$$\nu_j^i \geq 0, \mu_j^i = 0 \Rightarrow (f_j^i - \gamma_i)y_j^i \leq 0.$$

We define the following index sets at a given $\boldsymbol{\alpha}$:

$$\begin{aligned}
I_0^i &= \{j|0 < \alpha_j^i < \frac{1}{n}\} \\
I_1^i &= \{j|y_j^i = 1, \alpha_j^i = 0\} \\
I_2^i &= \{j|y_j^i = -1, \alpha_j^i = \frac{1}{n}\} \\
I_3^i &= \{j|y_j^i = 1, \alpha_j^i = \frac{1}{n}\} \\
I_4^i &= \{j|y_j^i = -1, \alpha_j^i = 0\}
\end{aligned}$$

Then we can get

$$\begin{aligned}
f_j^i &\geq \gamma_i \qquad \forall j \in I_0^i \cup I_1^i \cup I_2^i \\
f_j^i &\leq \gamma_i \qquad \forall j \in I_0^i \cup I_3^i \cup I_4^i.
\end{aligned}$$

We next define c^i_j and d^i_j as

$$c^i_j = \begin{cases} 0 & \text{if } y^i_j = 1 \\ -\frac{1}{n} & \text{if } y^i_j = -1 \end{cases}$$
$$d^i_j = \begin{cases} \frac{1}{n} & \text{if } y^i_j = 1 \\ 0 & \text{if } y^i_j = -1 \end{cases}.$$

Then we can get $c^i_j \leq y^i_j \alpha^i_j \leq d^i_j$ for any i and j. Let $A_i = \{j | y^i_j \alpha^i_j < d^i_j\}$ and $B_i = \{j | y^i_j \alpha^i_j > c^i_j\}$. It is easy to verify that $A_i = I^i_0 \cup I^i_1 \cup I^i_2$ and $B_i = I^i_0 \cup I^i_3 \cup I^i_4$. So we can get

$$\begin{aligned} f^i_j &\geq \gamma_i \qquad \forall j \in A_i \\ f^i_j &\leq \gamma_i \qquad \forall j \in B_i. \end{aligned}$$

Let $F^i_{up} = \min\{f^i_j | j \in A_i\}$ and $F^i_{low} = \max\{f^i_j | j \in B_i\}$. Then $\boldsymbol{\alpha}$ is the optimal solution of problem (10) if and only if $F^i_{up} \geq F^i_{low}$ for $i = 1, \ldots, m$.

So if at present $\boldsymbol{\alpha}$ is not the optimal solution, then it means there exist $j \in A_i$ and $k \in B_i$ such that $f^i_j < f^i_k$. Then (i, j, k) is called a 'violating tuple'.

We use an SMO-style algorithm to optimize problem (10). The core of this algorithm is to define the criterion to select the working set. By using the first-order information, we can select the working set via the 'most violating tuple' (i, j, k): $j = \arg\min_t \{f^i_t | t \in A_i\}$ and $k = \arg\max_t \{f^i_t | t \in B_i\}$ for $i = 1, \ldots, m$.

Here we use the second-order information to select the working set, which is shown to be more effective than using the first-order information in standard SVM [Fan et al. 2005]. Let the current estimate be denoted as $\boldsymbol{\alpha}$ and we want to update the parameters for the ith task. Then we want to update $\boldsymbol{\alpha}$ by an incremental value $\mathbf{d}$ which has only two nonzero elements. Let us denote the objective function of problem (10) by $h(\boldsymbol{\alpha})$. Since $h(\boldsymbol{\alpha})$ is quadratic,

$$\begin{aligned} h(\boldsymbol{\alpha} + \mathbf{d}) - h(\boldsymbol{\alpha}) &= \nabla h(\boldsymbol{\alpha})^T \mathbf{d} + \frac{1}{2} \mathbf{d}^T \nabla^2 h(\boldsymbol{\alpha}) \mathbf{d} \\ &= (\nabla h(\boldsymbol{\alpha})^i_B)^T \mathbf{d}^i_B + \frac{1}{2} (\mathbf{d}^i_B)^T \nabla^2 h(\boldsymbol{\alpha})^i_{BB} \mathbf{d}^i_B, \end{aligned}$$

where B denotes the indices of the nonzero elements. Problem (10) is equivalent to the following problem by using second-order information:

$$\min_{B:|B|=2} Sub(B), \tag{29}$$

where

$$\begin{aligned} Sub(B) \equiv \min_{\mathbf{d}^i_B} \quad & \frac{1}{2} (\mathbf{d}^i_B)^T \nabla^2 h(\boldsymbol{\alpha})^i_{BB} \mathbf{d}^i_B + (\nabla h(\boldsymbol{\alpha})^i_B)^T \mathbf{d}^i_B \\ \text{s.t.} \quad & \mathbf{y}^i_B \mathbf{d}^i_B = 0 \\ & d^i_t \geq 0, \ \text{if } \alpha^i_t = 0 \text{ and } t \in B \\ & d^i_t \leq 0, \ \text{if } \alpha^i_t = \frac{1}{n} \text{ and } t \in B. \end{aligned} \tag{30}$$

Note that problem (30) is to choose the working set B and so the constraint $0 \leq \alpha^i_j + d^i_j \leq \frac{1}{n_i}$ does not need to be satisfied. If problem (30) is solved directly, we have to enumerate

all possible situations which will cost $O(n^2)$ time. Here we use a greedy method instead:
1) Select j as $j = \arg\max_t\{f^i_j | j \in B_i\}$;
2) Consider $Sub(B)$ defined in (29) and select

$$k = \arg\min_t\{Sub(\{j,t\}) | t \in A_i, f^i_t < f^i_j\}. \tag{31}$$

This procedure costs $O(n)$ time. It is easy to show the relationship between f^i_j and $\nabla h(\boldsymbol{\alpha})^i_j$ as $f^i_j = y^i_j \nabla h(\boldsymbol{\alpha})^i_j$. Next we will show how to solve problem (31).

Let (i,j,k) be a violating tuple. Define $\hat{d}^i_j = y^i_j d^i_j$ and $\hat{d}^i_k = y^i_k d^i_k$. From $(\mathbf{y}^i_B)^T \mathbf{d}^i_B = 0$, we get $\hat{d}^i_j = -\hat{d}^i_k$. Then the objective function of $Sub(\{j,k\})$ becomes

$$\begin{aligned} & \frac{1}{2}[d^i_j \; d^i_k] \begin{bmatrix} \tilde{K}^i_{jj} & \tilde{K}^i_{jk} \\ \tilde{K}^i_{jk} & \tilde{K}^i_{kk} \end{bmatrix} \begin{bmatrix} d^i_j \\ d^i_k \end{bmatrix} + [\nabla h(\boldsymbol{\alpha})^i_j \; \nabla h(\boldsymbol{\alpha})^i_k] \begin{bmatrix} d^i_j \\ d^i_k \end{bmatrix} \\ = \; & \frac{1}{2}(K^i_{jj} + K^i_{kk} - 2K^i_{jk})(\hat{d}^i_k)^2 + (-f^i_j + f^i_k)\hat{d}^i_k, \end{aligned} \tag{32}$$

where $K^i_{jk} = k_{ML}((\mathbf{x}_j, i), (\mathbf{x}_k, i))$. We first assume $K^i_{jj} + K^i_{kk} - 2K^i_{jk} > 0$, then we can define $a^i_{jk} = K^i_{jj} + K^i_{kk} - 2K^i_{jk} > 0$ and $b^i_{jk} = -f^i_j + f^i_k < 0$ because of the definition of a violating pair. Then problem (32) obtains its minimum at

$$\hat{d}^i_k = -\hat{d}^i_j = -\frac{b^i_{jk}}{a^i_{jk}} > 0$$

and the objective function value equals $-\frac{(b^i_{jk})^2}{2a^i_{jk}}$. It is easy to verify that $\hat{d}^i_k$ and $-\hat{d}^i_j$ satisfy the remaining constraints of problem (30). When $k(\cdot,\cdot)$ is a positive definite kernel, $K^i_{jj} + K^i_{kk} - 2K^i_{jk} > 0$ holds for any i, j, k. Otherwise, problem (30) is changed to

$$\begin{aligned} Sub(B) \equiv \min_{\mathbf{d}^i_B} \quad & \frac{1}{2}(\mathbf{d}^i_B)^T \nabla^2 h(\boldsymbol{\alpha})^i_{BB} \mathbf{d}^i_B + (\nabla h(\boldsymbol{\alpha})^i_B)^T \mathbf{d}^i_B + \frac{\tau - a^i_{jk}}{4}((d^i_j)^2 + (d^i_k)^2) \\ \text{s.t.} \quad & \mathbf{y}^i_B \mathbf{d}^i_B = 0 \\ & d^i_t \geq 0, \text{ if } \alpha^i_t = 0 \text{ and } t \in B \\ & d^i_t \leq 0, \text{ if } \alpha^i_t = \frac{1}{n} \text{ and } t \in B, \end{aligned} \tag{33}$$

where τ is a small positive number. The optimal value of problem (33) is $-\frac{(b^i_{jk})^2}{2\tau}$. So the complete working set selection procedure is given as follows:
1) Calculate a^i_{jk} and b^i_{jk}, and define

$$\hat{a}^i_{jk} = \begin{cases} a^i_{jk} & \text{if } a^i_{jk} > 0 \\ \tau & \text{otherwise.} \end{cases}$$

2) Select $j = \arg\max_t\{f^i_j | j \in B_i\}$.
3) Select $k = \arg\min\{-\frac{(b^i_{jt})^2}{\hat{a}^i_{jt}} | t \in A_i, f^i_t < f^i_j\}$.

After working set selection, we need to update the model parameters. Let us define new variables $\tilde{\alpha}_j^i = y_j^i \alpha_j^i$. So problem (10) will become

$$\begin{aligned}
\min_{\tilde{\boldsymbol{\alpha}}} \quad & h(\tilde{\boldsymbol{\alpha}}) = \frac{1}{2}\tilde{\boldsymbol{\alpha}}^T \mathbf{K}_{ML}\tilde{\boldsymbol{\alpha}} - \mathbf{y}^T\tilde{\boldsymbol{\alpha}} \\
\text{s.t.} \quad & \sum_{j=1}^{n_i} \tilde{\alpha}_j^i = 0, \forall i \\
& c_j^i \le \tilde{\alpha}_j^i \le d_j^i, \forall i, j.
\end{aligned} \tag{34}$$

Suppose (i, j, k) defines a violation at some $\tilde{\boldsymbol{\alpha}}$. So we can adjust $\tilde{\alpha}_j^i$ and α_k^i to achieve an increase in f while maintaining the equality constraints $\sum_j \tilde{\alpha}_j^i = 0$ for $i = 1, \ldots, m$. We define the following update:

$$\begin{aligned}
&\tilde{\alpha}_j^i(t) = \tilde{\alpha}_j^i - t; \\
&\tilde{\alpha}_k^i(t) = \tilde{\alpha}_k^i + t; \\
&\text{other elements in } \tilde{\boldsymbol{\alpha}} \text{ remain fixed.}
\end{aligned}$$

The updated $\tilde{\boldsymbol{\alpha}}$ is denoted by $\tilde{\boldsymbol{\alpha}}(t)$. We define $\phi(t) = h(\tilde{\boldsymbol{\alpha}}(t))$ and minimize $\phi(t)$ to find the optimal $t^\star$. Since $\phi(t)$ is a quadratic function of t, $\phi(t) = \phi(0) + t\phi'(0) + \frac{t^2}{2}\phi''(0)$. It is easy to show that

$$\begin{aligned}
\phi'(t) &= \frac{\partial \phi(t)}{\partial t} \\
&= \frac{\partial \phi(t)}{\partial \tilde{\alpha}_j^i(t)} \frac{\partial \tilde{\alpha}_j^i(t)}{\partial t} + \frac{\partial \phi(t)}{\partial \tilde{\alpha}_k^i(t)} \frac{\partial \tilde{\alpha}_k^i(t)}{\partial t} \\
&= f_k^i(t) - f_j^i(t) \\
&= b_{jk}^i \\
\phi''(t) &= \frac{\partial \phi'(t)}{\partial t} \\
&= \frac{\partial \phi'(t)}{\partial \tilde{\alpha}_j^i(t)} \frac{\partial \tilde{\alpha}_j^i(t)}{\partial t} + \frac{\partial \phi'(t)}{\partial \tilde{\alpha}_k^i(t)} \frac{\partial \tilde{\alpha}_k^i(t)}{\partial t} \\
&= k_{ML}((\mathbf{x}_j, i), (\mathbf{x}_j, i)) + k_{ML}((\mathbf{x}_k, i), (\mathbf{x}_k, i)) - 2k_{ML}((\mathbf{x}_j, i), (\mathbf{x}_k, i)), \\
&= a_{jk}^i,
\end{aligned}$$

where $f_j^i(t)$ is the value of f_j^i at $\tilde{\boldsymbol{\alpha}}(t)$. Since $\tilde{\alpha}_j^i(t)$ and $\tilde{\alpha}_j^i(t)$ need to satisfy the constraints in problem (34), t needs to satisfy

$$t_1 \le t \le t_2,$$

where $t_1 = \max(c_k^i - \tilde{\alpha}_k^i, \tilde{\alpha}_j^i - d_j^i)$ and $t_2 = \min(d_k^i - \tilde{\alpha}_k^i, \tilde{\alpha}_j^i - c_j^i)$. So the optimal $t^\star$ can be calculated as

$$t^\star = \max(t_1, \min(t_2, -\frac{b_{jk}^i}{a_{jk}^i})). \tag{35}$$

After updating $\boldsymbol{\alpha}$, we can update f_q^p for all p, q as:

$$(f_q^p)^{new} = f_q^p + k_{ML}((\mathbf{x}_j, i), (\mathbf{x}_q, p))[\tilde{\alpha}_j^i(t^\star) - \tilde{\alpha}_j^i] + k_{ML}((\mathbf{x}_k, i), (\mathbf{x}_q, p))[\tilde{\alpha}_k^i(t^\star) - \tilde{\alpha}_k^i]. \tag{36}$$

Note that Eq. (35) only holds when $a^i_{jk} > 0$ since the optimization problem is convex. When $a^i_{jk} \leq 0$, similar to [Fan et al. 2005], the objective function we solved is changed to

$$h(\tilde{\boldsymbol{\alpha}}(t)) + \frac{\tau - a^i_{jk}}{2} t^2,$$

where τ is the parameter in the working set selection procedure. Then the optimal $t^\star$ is

$$t^\star = \max(t_1, \min(t_2, -\frac{b^i_{jk}}{\tau})). \tag{37}$$

Since it is usually not easy to achieve optimality exactly in numerical solutions, there is a need to define approximate optimality conditions. Here we use conditions similar to those in LIBSVM:

$$F^i_{low} \leq F^i_{up} + \varepsilon \text{ for } i = 1, \ldots, m,$$

where ε is a user-defined threshold.

Moreover, we can use the shrinking technique [Joachims 1998] to speed up the training procedure.

Appendix B

In this section, we give the proof of Theorem 3.

Proof of Theorem 3:

We first present two useful equations here:

$$\text{tr}(\mathbf{ABC}) = (\text{vec}(\mathbf{C}^T))^T(\mathbf{B}^T \otimes \mathbf{I}_a)\text{vec}(\mathbf{A}), \tag{38}$$

where $\otimes$ denotes the Kronecker product, $\mathbf{A} \in \mathbb{R}^{a\times b}$, $\mathbf{B} \in \mathbb{R}^{b\times c}$ and $\mathbf{C} \in \mathbb{R}^{c\times a}$, and

$$\text{tr}(\mathbf{ABCD}) = \text{vec}(\mathbf{A}^T)^T(\mathbf{D}^T \otimes \mathbf{B})\text{vec}(\mathbf{C}), \tag{39}$$

where $\mathbf{A} \in \mathbb{R}^{a\times b}$, $\mathbf{B} \in \mathbb{R}^{b\times c}$, $\mathbf{C} \in \mathbb{R}^{c\times d}$ and $\mathbf{D} \in \mathbb{R}^{d\times a}$. The proofs for these two equations can be found in [Seber 2007]. By using Eq. (39), the first term in the objective function of problem (18) can be reformulated as

$$\begin{aligned}
&\frac{1}{2\lambda_1}\text{tr}\Big(\Big(\tilde{\mathbf{Y}} \odot \begin{pmatrix}\boldsymbol{\Theta}\\ \boldsymbol{\Gamma}\end{pmatrix}\Big)^T \mathbf{K}\Big(\tilde{\mathbf{Y}} \odot \begin{pmatrix}\boldsymbol{\Theta}\\ \boldsymbol{\Gamma}\end{pmatrix}\Big)\boldsymbol{\Omega}\Big)\\
=&\frac{1}{2\lambda_1}\text{vec}\Big(\tilde{\mathbf{Y}} \odot \begin{pmatrix}\boldsymbol{\Theta}\\ \boldsymbol{\Gamma}\end{pmatrix}\Big)^T(\boldsymbol{\Omega} \otimes \mathbf{K})\text{vec}\Big(\tilde{\mathbf{Y}} \odot \begin{pmatrix}\boldsymbol{\Theta}\\ \boldsymbol{\Gamma}\end{pmatrix}\Big)\\
=&\frac{1}{2\lambda_1}\text{vec}\Big(\begin{pmatrix}\boldsymbol{\Theta}\\ \boldsymbol{\Gamma}\end{pmatrix}\Big)^T\text{diag}(\text{vec}(\tilde{\mathbf{Y}}))(\boldsymbol{\Omega} \otimes \mathbf{K})\text{diag}(\text{vec}(\tilde{\mathbf{Y}}))\text{vec}\Big(\begin{pmatrix}\boldsymbol{\Theta}\\ \boldsymbol{\Gamma}\end{pmatrix}\Big),
\end{aligned}$$

which is a quadratic function with respect to $\text{vec}\Big(\begin{pmatrix}\boldsymbol{\Theta}\\ \boldsymbol{\Gamma}\end{pmatrix}\Big)$. Here $\text{diag}(\cdot)$ is an operator that converts a vector to a diagonal matrix and the last equality holds because of the fact that $\text{vec}(\mathbf{A} \odot \mathbf{B}) = \text{diag}(\text{vec}(\mathbf{A}))\text{vec}(\mathbf{B})$ for matrices $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{a\times b}$. Since $\boldsymbol{\Omega}$ and $\mathbf{K}$ are PSD matrices, the first term in the objective function of problem (18) is convex with respect to $\text{vec}\Big(\begin{pmatrix}\boldsymbol{\Theta}\\ \boldsymbol{\Gamma}\end{pmatrix}\Big)$ due to a property of the Kronecker product that $\mathbf{A} \otimes \mathbf{B}$ is PSD when $\mathbf{A}$ and $\mathbf{B}$ are PSD matrices.

By using Eq. (38), the second term in the objective function of problem (18) can be rewritten as

$$\frac{1}{2\lambda_2}\mathrm{tr}(\mathbf{\Gamma}\mathbf{\Omega}\mathbf{\Gamma}^T) = \frac{1}{2\lambda_2}(\mathrm{vec}(\mathbf{\Gamma}))^T(\mathbf{\Omega}\otimes\mathbf{I}_u)\mathrm{vec}(\mathbf{\Gamma}),$$

which is also a quadratic function with respect to $\mathrm{vec}\left(\begin{pmatrix}\mathbf{\Theta}\\ \mathbf{\Gamma}\end{pmatrix}\right)$. Also due to the fact that $\mathbf{\Omega}$ is a PSD matrix, the second term of the objective function in problem (18) is also a convex function with respect to $\mathrm{vec}\left(\begin{pmatrix}\mathbf{\Theta}\\ \mathbf{\Gamma}\end{pmatrix}\right)$.

Since the third term in the objective function of problem (18) is linear and the constraints are all linear with respect to $\mathrm{vec}\left(\begin{pmatrix}\mathbf{\Theta}\\ \mathbf{\Gamma}\end{pmatrix}\right)$, problem (18) is convex with respect to $\mathrm{vec}\left(\begin{pmatrix}\mathbf{\Theta}\\ \mathbf{\Gamma}\end{pmatrix}\right)$, e.g., a QP problem. □