Regularized Matrix Factorization for Multilabel Learning With Missing Labels

Lei Feng, Jun Huang, Senlin Shu, and Bo An

***Abstract*—This article tackles the problem of multilabel learning with missing labels. For this problem, it is widely accepted that label correlations can be used to recover the ground-truth label matrix. Most of the existing approaches impose the low-rank assumption on the observed label matrix to exploit label correlations by decomposing it into two matrices, which describe the latent factors of instances and labels, respectively. The quality of these latent factors highly influences the recovery of ground-truth labels and the construction of the multilabel classification model. In this article, we propose recovering the ground-truth label matrix by regularized matrix factorization. Specifically, the latent factors of instances are regularized by the local topological structure derived from the feature space, which can be further used to induce an effective multilabel model. Moreover, the latent factors of labels and the label correlations are mutually adapted via label manifold regularization. In this way, the recovery of the ground-truth label matrix and the construction of the multilabel classification model are optimized jointly and can benefit from the regularized matrix factorization. Extensive experimental studies show that the proposed approach significantly outperforms the state-of-the-art algorithms on both full-label and missing-label data.**

***Index Terms*—Latent factors, multilabel learning, regularized matrix factorization.**

I. Introduction

In MULTILABEL learning [1]–[5], each instance is associated with multiple labels simultaneously. During the last few decades, multilabel learning has been widely applied in numerous real-world application domains [1], [6], [7], including image annotation [8], [9]; text categorization [10]; and object recognition [11]; Web mining [12]; bioinformatics [13]; etc.

Manuscript received December 25, 2019; revised June 7, 2020; accepted August 5, 2020. Date of publication September 16, 2020; date of current version May 19, 2022. This work was supported in part by the National Research Foundation, Singapore, through its AI Singapore Programme (AISG) under Award AISG-RP-2019-0013; in part by the National Satellite of Excellence in Trustworthy Software Systems under Award NSOE-TSS2019-01; and in part by NTU. This article was recommended by Associate Editor S. Cruces. *(Corresponding authors: Lei Feng; Jun Huang.)*

Lei Feng and Bo An are with the School of Computer Science and Engineering, Nanyang Technological University, Singapore 639798 (e-mail: feng0093@e.ntu.edu.sg; boan@ntu.edu.sg).

Jun Huang is with the School of Computer Science and Technology, Anhui University of Technology, Ma'anshan 243002, China (e-mail: huangjun.cs@ahut.edu.cn).

Senlin Shu is with the College of Computer and Information Science, Southwest University, Chongqing 400715, China (e-mail: ssl2018@email.swu.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TCYB.2020.3016897.

Digital Object Identifier 10.1109/TCYB.2020.3016897

In traditional multilabel learning studies, a basic assumption is that all the relevant labels of each training instance are known. However, in many applications, such an assumption is hard to hold because obtaining all relevant labels is difficult, and generally, only a partial label set can be observed. For example, many image annotation tasks use crowdsourcing platforms to collect labels. For each image, these labels are normally generated by collapsing annotated words from multiple users, while each user may sometimes ignore labels they do not know or have little interest.

The problem of multilabel learning with missing labels (MLMLs) undoubtedly increases the difficulty on training a robust multilabel classifier, and traditional multilabel learning techniques may be poor at addressing it [14]. There have been many attempts to recover the ground-truth label matrix by exploiting label correlations, thereby training an effective multilabel learning model [2], [15]–[19]. In [15], the formal definition of *weak label learning* [15] is initially proposed to deal with the incomplete relevant label set for multilabel learning, and the label correlations are exploited by assuming that there is a group of low-rank base similarities. Recently, there has been increasing interest in exploiting label correlations using the low-rank assumption on the label matrix, for example, matrix completion with side information [20], [21]; nuclear norm regularization [16], [18]; and matrix factorization [2], [17]. Under the assumption of low rank, the observed label matrix is generally decomposed into two matrices, which describe the latent factors of instances and labels, respectively. The well-learned latent factors of instances and labels should effectively guide the recovery of the ground-truth label matrix and the learning of the multilabel classification model. Intuitively, the local topological structure of the latent factors of instances is supposed to be consistent with that of the feature space, and the manifold of the latent factors of labels is supposed to be consistent with the label correlations.

Motivated by the above observations, in this article, we propose a novel method called RMFL, that is, regularized matrix factorization for MLMLs. To deal with missing labels, we suggest that the observed label matrix can be decomposed into two matrices, which describe the latent factors of instances and labels, respectively. On the one hand, the latent factors of instances are regularized by the local topological structural information in feature space. On the other hand, the latent factors of labels and the label correlations are mutually adapted via label manifold regularization; thus, the label correlations are exploited in an explicit manner. Such a regularized matrix factorization is further incorporated into model

training, which brings the advantage that the ground-truth label matrix is recovered while the desired multilabel learning model is trained simultaneously. Extensive experiments for learning with full labels and learning with missing labels show that RMFL significantly outperforms the state-of-the-art algorithms.

II. Related Work

Label embedding methods [22]–[28] have emerged as a mainstream solution for general multilabel learning (i.e., multilabel learning with full labels). These methods normally adopt different manipulations on the original label space to find an optimal low-dimensional latent label space and train the desired multilabel model from such latent label space for improving performance. For example, techniques, such as principal component analysis [29]; canonical correlation analysis [22]; manifold deduction [30], [31]; and sparse reconstruction [32], are used to transform the original label space to the desired latent label space. However, most of these label embedding methods are designed for general multilabel learning with full labels, which cannot deal with missing labels directly.

The problem of MLMLs has attracted great interest and many algorithms have been proposed to solve it. As label correlations are crucially important for multilabel learning, most of the existing algorithms aim to recover the ground-truth label matrix by exploiting label correlations with the low-rank assumption on the label matrix. Examples include matrix completion with side information [20], [21]; nuclear norm regularization [16], [18]; and matrix factorization [2], [17]. Although label correlations can be exploited via low-rank assumption, it would be more beneficial to exploit label correlations explicitly. In addition, the above approaches normally exploit label correlations by focusing on the label space and, thus, ignore making full use of potentially useful information in the feature space. One recent work [2] learns a latent label representation with both local and global label correlations to deal with missing labels. However, they only exploit the labeling information from the label space, while the potentially useful information in the feature space is not fully exploited. In the presence of missing labels, the label space could be noisy, and the learned label correlations may be misleading. Hence, how to effectively leverage the feature space to recover the label space becomes significantly important.

In the next section, a novel approach called RMFL based on regularized matrix factorization will be introduced for MLMLs. Different from the above approaches, RMFL not only leverages the topological information in feature space but also explicitly exploits the label correlations.

III. Proposed Approach

In MLMLs, the given label matrix is partially observed. Some approaches [15], [33] directly treat the missing labels as negative labels and deal with such label bias. In this article, we adopt the general setting [2], [17], [20] that both positive and negative labels can be missing. In other words, the observed label matrix is denoted by $\boldsymbol{Y} = [\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_m]^\top \in \{+1, 0, -1\}^{m\times l}$, where m denotes the number of examples, l denotes the number of labels, $+1$ denotes positive labels, -1 denotes negative labels, and 0 denotes missing labels. In addition, we denote by Ω the set containing indices of observed labels in $\boldsymbol{Y}$ (indices of nonzero elements in $\boldsymbol{Y}$), thus $[\Pi_\Omega(\boldsymbol{Y})]_{i,j}$ equals $\boldsymbol{Y}_{i,j}$ if $(i,j) \in \Omega$, and 0 otherwise. We denote by $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m]^\top \in \mathbb{R}^{m\times n}$ the matrix containing all the instances, where m is the number of instances and n is the number of dimensions of each instance. With these notations, we introduce RMFL in the following sections.

A. Regularized Matrix Factorization

1) Matrix Factorization: In multilabel learning, labels are normally correlated; thus, the label matrix is usually assumed to be low rank [2], [17], [18]. We assume that $\boldsymbol{Y}$ can be approximately represented by the product of two matrices

$$\min_{\boldsymbol{U},\boldsymbol{V}} \left\| \Pi_\Omega\left(\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{V}^\top\right)\right\|_F^2 \tag{1}$$

where $\|\cdot\|_F$ denotes the Frobenius norm, and $\boldsymbol{U} \in \mathbb{R}^{m\times k}$ and $\boldsymbol{V} \in \mathbb{R}^{l\times k}$ denote the latent factors (high level and abstract descriptions) of instances and labels, respectively. Note that with the presence of Π_Ω, we only focus on the positive and negative labels, and missing labels would not incur any loss. However, such decomposition fails to leverage potential useful information in feature space and correlations among labels, to regularize the latent factors of instances and labels.

2) Regularization on Instances: To regularize the latent factors of instances, we suggest that the topological structure of the feature space can be transferred to the latent factors of instance local by local, which means, only local topological structure of the feature space can be transferred. In order to retain such locality, we need to first use the local neighborhood information of each instance to construct a KNN graph and denote by $\mathcal{N}_i$ the set of K-nearest neighbors of the instance $\boldsymbol{x}_i$. Then, we assume that each instance can be optimally reconstructed by a linear combination of its neighbors [34]; thus, the reconstruction weights are computed as follows:

$$\begin{aligned} \min_{\boldsymbol{S}} \ & \sum_{i=1}^{m} \left\| \boldsymbol{x}_i - \sum_{\boldsymbol{x}_j \in \mathcal{N}_i} s_{ij}\boldsymbol{x}_j \right\|_2^2 \\ \text{s.t.} \ & \sum_j s_{ij} = 1 \ \forall \ 1 \le i \le m \end{aligned} \tag{2}$$

where $\boldsymbol{S}$ is the weight matrix and $s_{ij} = 0$ if $\boldsymbol{x}_j \notin \mathcal{N}_i$. Problem (2) can be solved by the following m independent standard quadratic programming problems:

$$\begin{aligned} \min_{\boldsymbol{s}_i} \ & \boldsymbol{s}_i^\top \boldsymbol{G}_i \boldsymbol{s}_i \\ \text{s.t.} \ & \boldsymbol{1}^\top \boldsymbol{s}_i = 1 \end{aligned} \tag{3}$$

where $\boldsymbol{G}_i$ is the local Gram matrix at $\boldsymbol{x}_i$ with $[\boldsymbol{G}_i]_{j,k} = (\boldsymbol{x}_i - \boldsymbol{x}_j)^\top(\boldsymbol{x}_i - \boldsymbol{x}_k)$ where $\boldsymbol{x}_j, \boldsymbol{x}_k \in \mathcal{N}_i$. With the transferred topological structure, the latent factors of instances can be regularized by

$$\min_{\boldsymbol{U}} \sum_{i=1}^{m} \left\| \boldsymbol{u}_i - \sum_j s_{ij}\boldsymbol{u}_j \right\|_2^2 = \min_{\boldsymbol{U}} \|\boldsymbol{U} - \boldsymbol{S}\boldsymbol{U}\|_F^2. \tag{4}$$

3) Regularization on Labels: To regularize the latent factors of labels, we suggest that the latent factors of similar labels should be similar. Concretely, let $\boldsymbol{C} = [c_{ij}]_{l\times l}$ be the label correlation matrix where c_{ij} denotes the similarity between the ith label and the jth label. We suggest that the distance $\|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2$ should be small with a high similarity c_{ij}. Thus, the label manifold regularization is presented as

$$\min_{\boldsymbol{V}} \sum_{i,j}^{l} c_{ij} \|\boldsymbol{v}_i - \boldsymbol{v}_j\|_2^2 = \min_{\boldsymbol{V}} \operatorname{tr}\left(\boldsymbol{V}^\top \boldsymbol{L} \boldsymbol{V}\right) \tag{5}$$

where $\operatorname{tr}(\cdot)$ is the trace operator, $\boldsymbol{L} = \operatorname{diag}(\boldsymbol{C}\mathbf{1}) - \boldsymbol{C}$ is the Laplacian matrix, and $\mathbf{1}$ denotes the vector with all components set to 1. Although there are multiple ways to calculate $\boldsymbol{C}$ with full labels, it could be much more difficult to calculate $\boldsymbol{C}$ with missing labels, since missing labels may severely ruin the label space. To overcome this problem, instead of specifying any correlation metric or label correlation matrix, we choose to learn the Laplacian matrix $\boldsymbol{L}$ directly. Specifically, we suggest that $\boldsymbol{L}$ and $\boldsymbol{V}$ are mutually adapted via label manifold regularization, thus label correlations are exploited in an explicit manner. Note that $\boldsymbol{L}$ is symmetric positive definite, hence we can decompose $\boldsymbol{L}$ into $\boldsymbol{Z}\boldsymbol{Z}^\top$ where $\boldsymbol{Z} \in R^{l\times k}$ [2]. To avoid the problem that optimization with respect to (w.r.t.) $\boldsymbol{Z}$ may lead to trivial solution $\boldsymbol{Z} = \mathbf{0}$, we add a constraint that each diagonal element of $\boldsymbol{Z}\boldsymbol{Z}^\top$ is set to 1. In this way, the following optimization problem is presented:

$$\begin{aligned} \min_{\boldsymbol{V},\boldsymbol{Z}} \ & \operatorname{tr}\left(\boldsymbol{V}^\top \boldsymbol{Z}\boldsymbol{Z}^\top \boldsymbol{V}\right) \\ \text{s.t.} \ & \boldsymbol{z}_i \boldsymbol{z}_i^\top = 1 \ \forall \ 1 \le i \le l \end{aligned} \tag{6}$$

where $\boldsymbol{z}_i$ is the ith row vector of $\boldsymbol{Z}$. It is worth noting that (6) comes from [2], and we apply it here for accurately capturing the label correlations.

Let $\boldsymbol{R} = [r_{ij}]_{m\times l}$ be the indicator matrix where r_{ij} equals 1 if $(i,j) \in \Omega$ and 0 otherwise. Thus, $\Pi_\Omega(\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{V}^\top)$ can be represented by the Hadamard product $\boldsymbol{R} \circ (\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{V}^\top)$. By combining (1), (3), and (5) with tradeoff parameters λ_1 and λ_2, we propose a novel regularized matrix factorization method as follows:

$$\begin{aligned} \min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{Z}} \ & \frac{1}{2}\left\|\boldsymbol{R} \circ \left(\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{V}^\top\right)\right\|_F^2 + \frac{\lambda_1}{2}\|\boldsymbol{U} - \boldsymbol{S}\boldsymbol{U}\|_F^2 \\ & + \frac{\lambda_2}{2}\operatorname{tr}\left(\boldsymbol{V}^\top \boldsymbol{Z}\boldsymbol{Z}^\top \boldsymbol{V}\right) \\ \text{s.t.} \ & \boldsymbol{z}_i \boldsymbol{z}_i^\top = 1 \ \forall \ 1 \le i \le l. \end{aligned} \tag{7}$$

B. Regularized Matrix Factorization for MLML

Since $\boldsymbol{U} \in \mathbb{R}^{m\times k}$ can be regarded as the latent labels (high level or abstract descriptions) of $\boldsymbol{X} \in \mathbb{R}^{m\times n}$, we can learn from such latent labels $\boldsymbol{U}$, thereby getting out of the dilemma of missing labels. To instantiate the learning model, we use the widely used squares loss $\|\boldsymbol{U} - f(\boldsymbol{X})\|_F^2$ where f is the common model: $f(\boldsymbol{X}) = \boldsymbol{X}\boldsymbol{W} + \mathbf{1}\boldsymbol{b}^\top$. In addition, we use the common squared Frobenius norm $\|\mathbf{W}\|_F^2$ to control the model complexity. By integrating the regularized matrix factorization into this learning model, we obtain the final objective function

$$\begin{aligned} \min_{\boldsymbol{U},\boldsymbol{V},\boldsymbol{Z},\boldsymbol{W},\boldsymbol{b}} \ & \frac{1}{2}\left\|\boldsymbol{R} \circ \left(\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{V}^\top\right)\right\|_F^2 + \frac{\lambda_1}{2}\|\boldsymbol{U} - \boldsymbol{S}\boldsymbol{U}\|_F^2 \\ & + \frac{\lambda_2}{2}\operatorname{tr}\left(\boldsymbol{V}^\top \boldsymbol{Z}\boldsymbol{Z}^\top \boldsymbol{V}\right) + \frac{\lambda_3}{2}\left\|\boldsymbol{U} - \boldsymbol{X}\boldsymbol{W} - \mathbf{1}\boldsymbol{b}^\top\right\|_F^2 \\ & + \frac{\lambda_4}{2}\|\boldsymbol{W}\|_F^2 \\ \text{s.t.} \ & \boldsymbol{z}_i \boldsymbol{z}_i^\top = 1 \ \forall \ 1 \le i \le l \end{aligned} \tag{8}$$

where λ_3 and λ_4 control the weight of the training loss incurred by the model f with the target $\mathbf{U}$ and the model complexity, respectively. Clearly, the proposed model tackles missing labels by matrix factorization, and the regularization terms encourage that similar labels have similar outputs, and structural information in feature space is also retained in output space.

IV. Optimization

Problem (8) can be solved by alternating optimization, which enables us to iteratively update each variable in an alternating way. Besides, the MANOPT toolbox [35] is employed to implement gradient descent with line search on the Euclidean space.

A. Updating $\boldsymbol{V}$

With $\boldsymbol{U}$, $\boldsymbol{Z}$, $\boldsymbol{W}$, and $\boldsymbol{b}$ fixed, problem (8) reduces to

$$\min_{\boldsymbol{V}} \frac{1}{2}\left\|\boldsymbol{R} \circ \left(\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{V}^\top\right)\right\|_F^2 + \frac{\lambda_2}{2}\operatorname{tr}\left(\boldsymbol{V}^\top \boldsymbol{Z}\boldsymbol{Z}^\top \boldsymbol{V}\right). \tag{9}$$

Computing the gradient for $\mathbf{V}$, we have

$$\nabla_{\boldsymbol{V}} = \left(\boldsymbol{R}^\top \circ \left(\boldsymbol{V}\boldsymbol{U}^\top - \boldsymbol{Y}^\top\right)\right)\boldsymbol{U} + \lambda_2 \boldsymbol{Z}\boldsymbol{Z}^\top \boldsymbol{V}. \tag{10}$$

B. Updating $\boldsymbol{W}$ and $\boldsymbol{b}$

With $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{Z}$ fixed, problem (8) reduces to

$$\min_{\boldsymbol{W},\boldsymbol{b}} \frac{\lambda_3}{2}\left\|\boldsymbol{U} - \boldsymbol{X}\boldsymbol{W} - \mathbf{1}\boldsymbol{b}^\top\right\|_F^2 + \frac{\lambda_4}{2}\|\boldsymbol{W}\|_F^2. \tag{11}$$

By setting $\hat{\lambda} = (\lambda_4/\lambda_3)$, the gradient w.r.t. $\boldsymbol{W}$ and $\boldsymbol{b}$ can be obtained as follows:

$$\begin{aligned} \nabla \boldsymbol{W} &= \boldsymbol{X}^\top\left(\boldsymbol{X}\boldsymbol{W} - \boldsymbol{U} + \mathbf{1}\boldsymbol{b}^\top\right) + \hat{\lambda}\boldsymbol{W} \\ \nabla \boldsymbol{b} &= \left(\boldsymbol{W}^\top \boldsymbol{X}^\top + \boldsymbol{b}\mathbf{1}^\top - \boldsymbol{U}^\top\right)\mathbf{1}. \end{aligned} \tag{12}$$

By further setting $\nabla_{\boldsymbol{W}}$ and $\nabla_{\boldsymbol{b}}$ to 0, it would be easy to obtain the closed-form solutions

$$\begin{aligned} \boldsymbol{W} &= \left(\boldsymbol{X}^\top \boldsymbol{X} + \hat{\lambda}\boldsymbol{I} - \frac{\boldsymbol{X}^\top \mathbf{1}\mathbf{1}^\top \boldsymbol{X}}{m}\right)^{-1}\left(\boldsymbol{X}^\top \boldsymbol{U} - \frac{\boldsymbol{X}^\top \mathbf{1}\mathbf{1}^\top \boldsymbol{U}}{m}\right) \\ \boldsymbol{b} &= \frac{1}{m}\left(\boldsymbol{U}^\top - \boldsymbol{W}^\top \boldsymbol{X}^\top\right)\mathbf{1}. \end{aligned} \tag{13}$$

Here, the learning model employed in (13) is a linear-in-input model, which cannot deal with the nonlinear case where the given data are not linearly separable. It is worth noting that many of the existing multilabel learning algorithms (e.g., [2], [36], and [37]) adopt the linear model, while we

Algorithm 1 RMFL Algorithm

Require:
the multi-label training set $\mathcal{D} = \{(\boldsymbol{X}, \boldsymbol{Y})\}$
the observation indicator matrix $\boldsymbol{R}$
the unseen test instance $\hat{\boldsymbol{x}}$
Ensure:
the predicted label vector $\hat{\boldsymbol{y}}$

learn the weight matrix $\boldsymbol{S}$ by solving problem (3);
construct the kernel matrix $\boldsymbol{K}$ by Gaussian kernel;
randomly initialize $\boldsymbol{U}$ and $\boldsymbol{Z}$;
repeat
  update $\boldsymbol{V}$ in terms of eq. (10);
  update $\boldsymbol{A}$ and $\boldsymbol{b}$ in terms of eq. (16);
  update $\boldsymbol{T} = \boldsymbol{K}\boldsymbol{A} + \mathbf{1}\boldsymbol{b}^\top$;
  update $\boldsymbol{U}$ in terms of eq. (18);
  update $\boldsymbol{Z}$ in terms of eqs. (20) and (21);
until convergence or the maximum number of iterations.
return $\hat{\boldsymbol{y}} = \text{sign}(\boldsymbol{V}(\sum_{i=1}^{m} \boldsymbol{a}_i \kappa(\hat{\boldsymbol{x}}, \boldsymbol{x}_i) + \boldsymbol{b}))$.

expect that better performance could be achieved if a nonlinear model is used. Therefore, we introduce the nonlinear kernel extension [38], [39]. Specifically, we utilize the feature mapping $\varphi(\cdot)$ to map the feature space $\boldsymbol{X}$ to some higher (maybe infinite) dimensional Hilbert space $\varphi(\boldsymbol{X})$. According to the representer theorem [40], $\boldsymbol{W}$ can be represented by a linear combination of input variables, that is, $\boldsymbol{W} = \varphi(\boldsymbol{X})^\top \boldsymbol{A}$ where $\boldsymbol{A} = [a_{ij}]_{m\times k}$ stores the weights. Let $\boldsymbol{K} = [k_{ij}]_{m\times m} = \varphi(\boldsymbol{X})\varphi(\boldsymbol{X})^\top$ be the kernel matrix with $k_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ where $\kappa(\cdot,\cdot)$ denotes the employed kernel function. In this article, we use the Gaussian kernel function

$$\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp\left(\frac{-\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2^2}{2\sigma^2}\right) \tag{14}$$

where σ is set to the average Euclidean distance of paired instances. In this way, $\varphi(\boldsymbol{X})\boldsymbol{W} = \varphi(\boldsymbol{X})\varphi(\boldsymbol{X})^\top \boldsymbol{A} = \boldsymbol{K}\boldsymbol{A}$, thus problem (11) can be written as

$$\min_{\boldsymbol{A},\boldsymbol{b}} \frac{1}{2}\left\|\boldsymbol{U} - \boldsymbol{K}\boldsymbol{A} - \mathbf{1}\boldsymbol{b}^\top\right\|_F^2 + \frac{\hat{\lambda}}{2}\text{tr}\left(\boldsymbol{A}^\top \boldsymbol{K}\boldsymbol{A}\right) \tag{15}$$

where we have used the property of the trace operator, that is, $\|\boldsymbol{W}\|_F^2 = \text{tr}(\boldsymbol{W}^\top\boldsymbol{W}) = \text{tr}(\boldsymbol{A}^\top\boldsymbol{K}\boldsymbol{A})$. Similarly, we can obtain the gradient w.r.t. $\boldsymbol{A}$ and $\boldsymbol{b}$ to $\mathbf{0}$ as follows:

$$\nabla_{\boldsymbol{A}} = \boldsymbol{K}^\top\left(\boldsymbol{K}\boldsymbol{A} - \boldsymbol{U} + \mathbf{1}\boldsymbol{b}^\top\right) + 2\hat{\lambda}\boldsymbol{A}$$
$$\nabla_{\boldsymbol{b}} = \left(\boldsymbol{A}^\top\boldsymbol{K}^\top + \boldsymbol{b}\mathbf{1}^\top - \boldsymbol{U}^\top\right)\mathbf{1}.$$

Hence, we can obtain the following closed-form solutions:

$$\boldsymbol{A} = \left(\boldsymbol{K}^\top\boldsymbol{K} + \hat{\lambda}\boldsymbol{K} - \frac{\boldsymbol{K}^\top\mathbf{1}\mathbf{1}^\top\boldsymbol{K}}{m}\right)^{-1}\left(\boldsymbol{K}^\top\boldsymbol{U} - \frac{\boldsymbol{K}^\top\mathbf{1}\mathbf{1}^\top\boldsymbol{U}}{m}\right)$$
$$\boldsymbol{b} = \frac{1}{m}\left(\boldsymbol{U}^\top - \boldsymbol{A}^\top\boldsymbol{K}^\top\right)\mathbf{1}. \tag{16}$$

C. Updating $\boldsymbol{U}$

With $\boldsymbol{V}$, $\boldsymbol{Z}$, $\boldsymbol{W}$, and $\boldsymbol{b}$ fixed, we denote by $\boldsymbol{T} = \varphi(\boldsymbol{X})\boldsymbol{W} + \mathbf{1}\boldsymbol{b}^\top = \boldsymbol{K}\boldsymbol{A} + \mathbf{1}\boldsymbol{b}^\top$, problem (8) reduces to

$$\min_{\boldsymbol{U}} \frac{1}{2}\left\|\boldsymbol{R}\circ\left(\boldsymbol{Y} - \boldsymbol{U}\boldsymbol{V}^\top\right)\right\|_F^2 + \frac{\lambda_1}{2}\|\boldsymbol{U} - \boldsymbol{S}\boldsymbol{U}\|_F^2 + \frac{\lambda_3}{2}\|\boldsymbol{U} - \boldsymbol{T}\|_F^2. \tag{17}$$

The gradient w.r.t. $\boldsymbol{U}$ can be obtained as follows:

$$\nabla_{\boldsymbol{U}} = \boldsymbol{R}\circ\left(\boldsymbol{U}\boldsymbol{V}^\top - \boldsymbol{Y}\right)\mathbf{V} + \lambda_1\left(\boldsymbol{I} + \boldsymbol{S}^\top\boldsymbol{S} - \boldsymbol{S}^\top - \boldsymbol{S}\right)\boldsymbol{U} + \lambda_3(\boldsymbol{U} - \boldsymbol{T}). \tag{18}$$

D. Updating $\boldsymbol{Z}$

With $\boldsymbol{U}$, $\boldsymbol{V}$, $\boldsymbol{W}$, and $\boldsymbol{b}$ fixed, problem (8) reduces to

$$\min_{\boldsymbol{Z}} \frac{1}{2}\text{tr}\left(\boldsymbol{V}^\top\boldsymbol{Z}\boldsymbol{Z}^\top\boldsymbol{V}\right) \quad \text{s.t. } \boldsymbol{z}_i\boldsymbol{z}_i^\top = 1 \ \forall\ 1 \le i \le l. \tag{19}$$

The gradient w.r.t. $\boldsymbol{Z}$ can be obtained as follows:

$$\nabla_{\boldsymbol{Z}} = \boldsymbol{V}\boldsymbol{V}^\top\boldsymbol{Z}. \tag{20}$$

To satisfy the constraint $\boldsymbol{z}_i\boldsymbol{z}_i^\top = 1$, we project each row of $\boldsymbol{Z}$ onto the unit ℓ_2 norm ball after each update

$$\boldsymbol{z}_i = \frac{\boldsymbol{z}_i}{\|\boldsymbol{z}_i\|_2} \ \forall\ 1 \le i \le l. \tag{21}$$

After the completion of the optimization process, given the test instance $\hat{\boldsymbol{x}}$, we first induce the corresponding latent factors $\hat{\boldsymbol{u}} = (\sum_{i=1}^{m} \boldsymbol{a}_i\kappa(\hat{\boldsymbol{x}}, \boldsymbol{x}_i) + \boldsymbol{b})$, and then derive the predicted label vector $\hat{\boldsymbol{y}} = \text{sign}(\boldsymbol{V}\hat{\boldsymbol{u}})$.

The pseudocode of RMFL is presented in Algorithm 1. The convergence of RMFL will be empirically demonstrated by experiments.

V. Experiments

In this section, we evaluate our proposed approach through extensive experiments.

A. Experimental Setup

In this section, we empirically evaluate our proposed approach by comparing with seven state-of-the-art algorithms on multiple real-world multilabel benchmark datasets, in terms of five widely used evaluation metrics.

1) Real-World Multilabel Benchmark Datasets: We conduct experiments on 30 benchmark datasets from various domains. Table I shows the detailed information of these datasets, including the number of instances, the number of dimensions for each instance, the number of classes, and the corresponding domain. For each dataset, we generate missing labels by considering the missing label ratio (M.L.Ratio) $\{40\%, 60\%, 80\%\}$. In other words, we randomly drop $\{\llcorner 0.4l \lrcorner, \llcorner 0.6l \lrcorner, \llcorner 0.8l \lrcorner\}$ labels for each instance in the training set. We compare all the algorithms under the same data setting. For each dataset, we randomly choose 80% examples to form the training set, and the remaining 20% examples

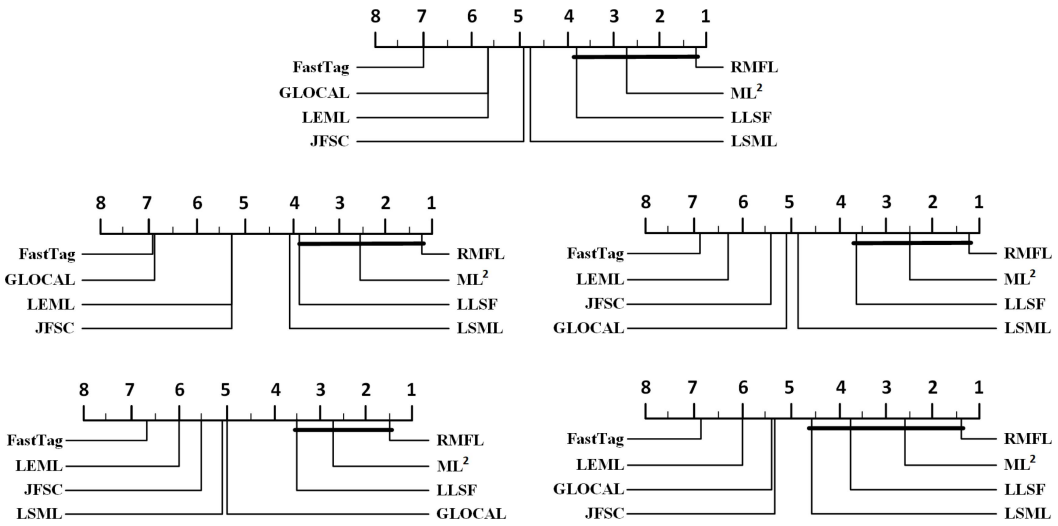


Fig. 1. Statistical comparison of RMFL (the control algorithm) against other comparing algorithms based on the *Nemenyi test* for learning with full labels. Algorithms not connected with RMFL in the CD diagram are considered to have a significantly different performance from the control algorithm.

TABLE I
DETAILED INFORMATION OF THE BENCHMARK DATASETS

Dataset	Instances	Dimensions	Labels	Domain
yeast	2417	103	14	biology
image	2000	294	5	image
scene	2407	294	6	image
emotions	593	72	6	music
corel5k	5000	499	374	image
science	5000	743	40	text
education	5000	550	33	text
social	5000	1047	39	text
pascal07	9963	512	20	image
corel16ks1	13766	500	153	image
corel16ks2	13761	500	164	image
corel16ks3	13760	500	154	image

constitute the test set. To reduce statistical variability, all the experimental results are averaged over ten independent repetitions, and mean results with standard deviations are reported.

2) Comparing Algorithms: We compare RMFL with seven state-of-the-art multilabel learning algorithms.

1) *FastTag [41]:* It reconstructs the ground-truth label matrix and learns the multilabel classification model simultaneously. Parameter *topk* is set to be the number of classes l for each dataset, and other parameters are automatically tuned by its built-in function.
2) *LEML [17]:* It exploits the low-rank assumption to construct the loss function for learning with missing labels. Parameters: p is set to the number of classes l for each dataset, k is selected in $\{0.1l, \ldots, 0.5l\}$, and the regularization parameter is selected in $\{10^{-5}, \ldots, 10^{-1}\}$.
3) *GLOCAL [2]:* It utilizes both global and local label correlations to perform MLMLs. Parameters: $\lambda = 1$, $\lambda_2 = 0.125$, and λ_3, λ_4 are searched in $\{10^{-6}, \ldots, 10^{0}\}$, and g is searched in $\{2^2, \ldots, 2^5\}$.
4) *LSML [36]:* It learns label-specific features for MLMLs. Parameters: λ_1, λ_2, λ_3, and λ_4 are searched in $\{10^{-5}, \ldots, 10^{3}\}$.
5) *ML² [30]:* It leverages the manifold in the label space for multilabel learning. Parameters: C_1 and C_2 are searched in $\{1, \ldots, 10\}$.
6) *JFSC [37]:* It performs joint feature selection and classification for multilabel learning. Parameters: α, β, and γ are searched in $\{4^{-5}, \ldots, 4^{5}\}$, and η is searched in $\{0.1, 1, 10\}$.
7) *LLSF [42]:* It learns label-specific features for multilabel learning. Parameters: α and β are searched in $\{2^{-10}, \ldots, 2^{10}\}$.
8) *RMFL:* This is our proposed approach, which leverages regularized matrix factorization for MLMLs. Parameters: λ_1 and λ_2 are searched in $\{10^{-5}, \ldots, 10^{5}\}$, λ_3 is selected from $\{1, 2\}$, and λ_4 is searched in $\{4^{-7}, \ldots, 4^{2}\}$.

In addition, for methods that use the *K*NN technique, K is empirically fixed at 10. For methods that use latent representations, the latent representation dimension k is simply set to 20. The parameters for all the above algorithms are selected by five-fold cross-validation on the training set. For the above approaches that use the *K*NN technique, K is empirically fixed at 10.

3) Evaluation Criteria: We measure the performance of each algorithm in items of five widely used evaluation criteria, including one error ($\mathcal{O}$), Hamming loss ($\mathcal{H}$), ranking loss ($\mathcal{R}$), coverage ($\mathcal{C}$), and average precision ($\mathcal{A}$). For average precision, the larger value means better performance ($\uparrow$). While for the other four criteria, the smaller the value, the better performance ($\downarrow$). More detailed information about these evaluation criteria is provided in [1].

B. Experimental Results

Here, we report the experimental results for learning with full labels and learning with missing labels.

1) Learning With Full Labels: Table II shows the predictive performance of each algorithm on fully labeled data (i.e., M.L.Ratio = 0). As can be seen, RMFL is better than

TABLE II
PREDICTION RESULTS FOR MULTILABEL LEARNING WITH FULL LABELS ON FIVE EVALUATION CRITERIA. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD, AND THE NUMBER IN THE BRACKET INDICATES THE RANKING OF THIS ALGORITHM. THE LAST COLUMN SHOWS THE AVERAGE RANKING OF EACH ALGORITHM ON EACH EVALUATION METRIC

		yeast	image	scene	emotions	corel5k	science	education	social	AvgR
$\mathcal{O}\downarrow$	RMFL	**.209±.010(1)**	**.251±.022(1)**	**.174±.015(1)**	.308±.018(2)	.632±.009(2)	**.475±.016(1)**	**.449±.018(1)**	**.264±.008(1)**	**1.250**
	ML^2	.213±.013(2)	.253±.020(2)	.175±.013(2)	.347±.048(3)	.653±.011(6)	.477±.014(2)	.457±.017(2)	.271±.011(3)	2.750
	LLSF	.360±.019(6)	.332±.023(4)	.254±.016(5)	.414±.032(4)	**.624±.011(1)**	.496±.018(4)	.460±.018(3)	.273±.008(4)	3.875
	JFSC	.220±.014(3)	.329±.026(3)	.343±.019(7)	.425±.037(5)	.647±.009(4)	.505±.011(6)	.476±.018(5)	.296±.008(6)	4.875
	FastTag	.247±.026(5)	.381±.046(7)	.240±.028(4)	.555±.065(8)	.694±.012(8)	.622±.031(8)	.593±.019(8)	.400±.025(8)	7.000
	LEML	.369±.018(8)	.448±.031(8)	.388±.015(8)	**.305±.038(1)**	.647±.010(4)	.495±.013(3)	.480±.018(6)	.316±.007(7)	5.625
	GLOCAL	.361±.015(7)	.337±.026(5)	.262±.013(6)	.457±.043(6)	.656±.009(7)	.498±.017(5)	.471±.015(4)	.284±.009(5)	5.625
	LSML	.220±.013(3)	.338±.021(6)	.231±.016(3)	.521±.023(7)	.640±.009(3)	.516±.016(7)	.501±.016(7)	.268±.012(2)	4.750
$\mathcal{H}\downarrow$	RMFL	**.189±.006(1)**	**.149±.008(1)**	**.073±.006(1)**	.218±.012(2)	**.008±.000(1)**	.032±.001(2)	**.036±.001(1)**	**.019±.000(1)**	**1.250**
	ML^2	.191±.007(2)	.151±.007(2)	.074±.005(2)	.239±.018(3)	.010±.000(4)	**.031±.001(1)**	.037±.001(2)	.020±.000(2)	2.250
	LLSF	.300±.004(6)	.201±.008(5)	.105±.002(4)	.264±.013(4)	.009±.000(2)	.034±.001(4)	.037±.001(2)	.020±.001(2)	3.625
	JFSC	.199±.004(3)	.199±.008(4)	.136±.003(7)	.282±.008(5)	.011±.000(7)	.035±.001(6)	.038±.001(4)	.021±.000(5)	5.125
	FastTag	.228±.007(5)	.215±.015(6)	.116±.013(5)	.529±.054(8)	.019±.001(8)	.056±.003(8)	.063±.004(8)	.032±.001(7)	6.250
	LEML	.426±.007(8)	.216±.006(7)	.134±.003(6)	**.206±.000(1)**	.010±.004(4)	.033±.001(3)	.040±.001(6)	.022±.001(6)	5.125
	GLOCAL	.303±.004(7)	.249±.005(8)	.179±.002(8)	.312±.007(7)	.009±.000(2)	.037±.000(7)	.044±.001(7)	.033±.000(8)	6.750
	LSML	.199±.004(3)	.188±.011(3)	.098±.004(3)	.299±.010(6)	.010±.000(4)	.034±.001(4)	.038±.001(4)	.020±.000(2)	3.625
$\mathcal{R}\downarrow$	RMFL	**.157±.005(1)**	**.134±.010(1)**	**.060±.006(1)**	.186±.012(2)	.138±.003(2)	**.114±.005(1)**	**.086±.004(1)**	**.064±.005(1)**	**1.250**
	ML^2	.159±.006(2)	.138±.008(2)	.062±.006(2)	.213±.031(3)	.172±.007(4)	.115±.007(2)	.087±.006(2)	.066±.004(2)	2.375
	LLSF	.341±.007(6)	.177±.015(4)	.089±.005(5)	.239±.017(4)	**.126±.004(1)**	.115±.005(2)	.087±.004(2)	.067±.005(3)	3.375
	JFSC	.170±.005(3)	.175±.016(3)	.119±.007(7)	.254±.023(5)	.177±.006(6)	.162±.007(7)	.115±.007(6)	.085±.007(6)	5.375
	FastTag	.185±.011(5)	.195±.019(7)	.082±.008(4)	.396±.037(8)	.220±.034(8)	.166±.014(8)	.153±.003(8)	.095±.008(8)	7.000
	LEML	.369±.007(8)	.230±.015(8)	.137±.004(8)	**.176±.013(1)**	.170±.006(5)	.142±.007(6)	.117±.006(7)	.090±.005(7)	6.250
	GLOCAL	.341±.007(6)	.179±.015(5)	.094±.005(6)	.265±.023(6)	.153±.006(3)	.134±.008(5)	.096±.005(4)	.081±.006(5)	5.000
	LSML	.170±.004(3)	.180±.014(6)	.081±.004(3)	.365±.021(7)	.177±.005(6)	.122±.007(4)	.102±.006(5)	.066±.004(3)	4.625
$\mathcal{C}\downarrow$	RMFL	**.442±.007(1)**	**.165±.013(1)**	**.064±.005(1)**	.313±.015(2)	.307±.004(2)	.160±.005(3)	**.127±.005(1)**	**.090±.006(1)**	**1.500**
	ML^2	.445±.006(2)	.167±.011(2)	.065±.006(2)	.336±.028(3)	.385±.075(6)	.159±.007(2)	.128±.007(2)	.092±.006(3)	2.750
	LLSF	.625±.008(8)	.196±.015(4)	.089±.004(5)	.355±.018(4)	**.281±.006(1)**	**.149±.005(1)**	.129±.004(3)	.091±.005(2)	3.500
	JFSC	.453±.007(3)	.195±.015(3)	.114±.006(7)	.372±.022(5)	.384±.009(5)	.214±.006(8)	.167±.008(7)	.119±.007(6)	5.500
	FastTag	.482±.015(5)	.211±.016(7)	.082±.007(3)	.502±.030(8)	.479±.022(8)	.210±.015(7)	.200±.010(8)	.124±.007(7)	6.625
	LEML	.621±.008(7)	.234±.013(8)	.127±.003(8)	**.303±.012(1)**	.379±.010(4)	.194±.007(6)	.164±.006(6)	.125±.005(8)	6.000
	GLOCAL	.615±.008(6)	.198±.016(5)	.093±.004(6)	.380±.012(6)	.339±.023(3)	.181±.008(5)	.140±.006(4)	.114±.007(5)	5.000
	LSML	.459±.006(4)	.203±.015(6)	.082±.004(3)	.447±.019(7)	.409±.009(7)	.179±.007(4)	.116±.006(5)	.087±.005(4)	5.000
$\mathcal{A}\uparrow$	RMFL	**.780±.007(1)**	**.836±.013(1)**	**.895±.008(1)**	.778±.012(2)	.301±.005(3)	**.613±.009(1)**	**.651±.013(1)**	**.784±.008(1)**	**1.375**
	ML^2	.778±.008(2)	.834±.010(2)	.897±.008(2)	.755±.029(3)	.289±.010(6)	.610±.009(2)	.645±.013(2)	.781±.009(2)	2.625
	LLSF	.617±.007(7)	.787±.015(4)	.847±.007(5)	.719±.017(4)	**.306±.008(1)**	.602±.010(3)	.643±.011(3)	.780±.008(3)	3.750
	JFSC	.761±.008(4)	.789±.016(3)	.793±.010(7)	.704±.021(5)	.290±.009(5)	.574±.008(7)	.623±.013(5)	.761±.008(6)	5.250
	FastTag	.742±.014(5)	.756±.027(7)	.856±.015(4)	.588±.033(8)	.238±.011(8)	.474±.023(7)	.508±.016(8)	.669±.017(8)	6.875
	LEML	.612±.009(8)	.722±.018(8)	.767±.008(8)	**.787±.019(1)**	.297±.007(4)	.591±.009(5)	.617±.014(7)	.736±.006(7)	6.000
	GLOCAL	.619±.007(6)	.784±.015(5)	.841±.007(6)	.689±.021(6)	.285±.006(7)	.592±.011(4)	.634±.011(4)	.766±.009(5)	5.375
	LSML	.764±.007(3)	.780±.014(6)	.861±.007(3)	.632±.011(7)	.303±.005(2)	.580±.011(6)	.618±.012(6)	.789±.010(4)	4.625

TABLE III
FRIEDMAN STATISTICS F_F ACCORDING TO EACH EVALUATION METRIC FOR LEARNING WITH FULL LABELS

	Evaluation metric	F_F	critical value (α = 0.05)
Full labels	One-error	8.49	2.203 (k = 8, N = 8)
	Hamming loss	12.43	
	Ranking loss	10.88	
	Coverage	7.41	
	Average precision	8.82	

other comparing algorithms on the whole. Concretely, on all datasets, across all the evaluation metrics, RMFL ranks first in 72.5% cases and ranks second in 22.5% cases. We can also observe that FastTag, LEML, LSML, and GLOCAL generally achieve inferior performance against other approaches for learning with full labels. This is due to that these four approaches are specially designed for learning with missing labels. Instead of extracting label correlations from the given label matrix, they aim to learn latent label correlations for recovering the given label matrix, thus useful labeling information is regrettably discarded when there are no missing labels. Although RMFL also aims to recover the given label matrix, the advantage of RMFL lies in that the proposed regularized matrix factorization not only enables similar labels to have similar predictions but also leverages the topological information in feature space for model training. However, we can also observe that RMFL obtains slightly worse results on the datasets *emotions* and *corel5k*. We speculate that this is because the alternating optimization method could get stuck in a local minimum, and this phenomenon is especially severe in these two datasets.

To further statistically perform performance comparison, the widely accepted *Friedman test* [43] is employed as the statistical test to analyze the relative performance among the comparing algorithms. Table III summarizes the Friedman statistics F_F and the corresponding critical value on each evaluation metric for learning with full labels. As shown in Table III, for learning with full labels, the null hypothesis that all the comparing algorithms perform equivalently is rejected on each evaluation metric at the significance level $\alpha = 0.05$. Furthermore, the *post-hoc Nemenyi test* [43] is employed to show the relative performance between our proposed algorithm (the control algorithm) and other comparing algorithms in detail. The significant differences between RMFL and other algorithms can be calibrated with the *critical difference* (CD). For learning with full labels, at the significance level $\alpha = 0.05$, we can obtain CD = 3.27 with the number of comparing algorithms $A = 8$ and the number of datasets $N = 8$. Fig. 1 illustrates the CD diagrams on each evaluation metric for

TABLE IV
PREDICTION RESULTS FOR MULTILABEL LEARNING WITH MISSING 40% LABELS ON FIVE EVALUATION CRITERIA. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD, AND THE NUMBER IN THE BRACKET INDICATES THE RANKING OF THIS ALGORITHM. THE LAST COLUMN SHOWS THE AVERAGE RANKING OF EACH ALGORITHM ON EACH EVALUATION METRIC

		yeast	image	scene	emotions	corel5k	science	education	social	AvgR
$\mathcal{O}\downarrow$	RMFL	**.221±.012(1)**	**.250±.017(1)**	**.180±.016(1)**	**.316±.017(1)**	**.653±.009(1)**	**.489±.016(1)**	**.455±.014(1)**	**.272±.009(1)**	**1.000**
	FastTag	.416±.033(5)	.381±.038(4)	.243±.018(3)	.596±.043(5)	.770±.019(5)	.638±.018(5)	.617±.023(5)	.436±.016(5)	4.625
	LEML	.357±.017(3)	.449±.028(5)	.388±.014(5)	.313±.040(2)	.660±.010(3)	.535±.010(4)	.499±.019(4)	.325±.003(4)	3.750
	GLOCAL	.367±.018(4)	.345±.027(3)	.269±.013(4)	.461±.041(3)	.661±.010(4)	.508±.014(3)	.474±.015(3)	.288±.011(3)	3.375
	LSML	.226±.018(2)	.337±.021(2)	.231±.015(2)	.534±.026(4)	.658±.010(2)	.489±.016(2)	.469±.017(2)	.274±.011(2)	2.250
$\mathcal{H}\downarrow$	RMFL	**.193±.005(1)**	**.149±.005(1)**	**.074±.005(1)**	.222±.008(2)	**.009±.000(1)**	**.031±.000(1)**	**.036±.000(1)**	**.019±.000(1)**	**1.250**
	FastTag	.243±.009(3)	.208±.016(3)	.115±.010(3)	.323±.056(5)	.015±.000(5)	.052±.003(5)	.061±.002(5)	.032±.001(4)	4.125
	LEML	.441±.008(5)	.215±.006(4)	.134±.003(4)	**.215±.010(1)**	**.009±.000(1)**	.043±.001(4)	.045±.001(4)	.024±.000(3)	3.375
	GLOCAL	.303±.004(4)	.249±.004(5)	.178±.001(5)	.311±.007(4)	**.009±.000(1)**	.036±.000(3)	.044±.000(3)	.032±.000(5)	3.875
	LSML	.220±.006(2)	.186±.011(2)	.097±.003(2)	.304±.009(3)	.010±.000(4)	.032±.000(2)	.037±.000(2)	.020±.000(2)	2.375
$\mathcal{R}\downarrow$	RMFL	**.164±.006(1)**	**.138±.009(1)**	**.059±.004(1)**	.190±.013(2)	**.150±.006(1)**	**.112±.005(1)**	**.081±.004(1)**	**.053±.004(1)**	**1.125**
	FastTag	.225±.010(3)	.193±.020(4)	.082±.010(3)	.399±.030(5)	.243±.006(5)	.166±.007(4)	.156±.011(5)	.104±.007(4)	4.125
	LEML	.359±.008(5)	.231±.015(5)	.136±.003(5)	**.178±.016(1)**	.181±.008(3)	.176±.007(5)	.141±.007(4)	.110±.007(5)	4.125
	GLOCAL	.341±.005(4)	.179±.014(2)	.094±.004(4)	.276±.020(3)	.167±.007(2)	.125±.007(3)	.090±.004(2)	.072±.005(3)	3.000
	LSML	.176±.005(2)	.179±.014(3)	.081±.004(2)	.361±.024(4)	.197±.006(4)	.124±.007(2)	.102±.007(3)	.066±.005(2)	2.625
$\mathcal{C}\downarrow$	RMFL	**.454±.008(1)**	**.168±.011(1)**	**.063±.005(1)**	.316±.016(2)	**.340±.008(1)**	**.158±.005(1)**	**.122±.005(1)**	**.085±.005(1)**	**1.125**
	FastTag	.455±.012(2)	.209±.015(4)	.083±.008(3)	.485±.036(5)	.461±.009(5)	.206±.009(4)	.196±.014(5)	.129±.008(4)	4.000
	LEML	.646±.009(5)	.236±.012(5)	.127±.003(5)	**.305±.016(1)**	.409±.016(3)	.234±.007(5)	.193±.007(4)	.150±.008(5)	4.125
	GLOCAL	.613±.008(4)	.197±.014(2)	.093±.003(4)	.390±.021(3)	.369±.014(2)	.170±.006(2)	.130±.006(2)	.102±.005(3)	2.750
	LSML	.472±.008(3)	.202±.015(3)	.082±.004(2)	.444±.023(4)	.455±.012(4)	.173±.006(3)	.150±.008(3)	.097±.006(2)	3.000
$\mathcal{A}\uparrow$	RMFL	**.769±.009(1)**	**.834±.010(1)**	**.893±.009(1)**	.774±.009(2)	**.290±.004(1)**	**.603±.011(1)**	**.646±.011(1)**	**.782±.007(1)**	**1.125**
	FastTag	.632±.017(3)	.756±.023(4)	.854±.013(3)	.567±.025(5)	.201±.005(5)	.467±.014(5)	.493±.018(5)	.642±.013(5)	4.375
	LEML	.612±.008(5)	.721±.016(5)	.767±.007(5)	**.784±.020(1)**	.283±.007(3)	.540±.006(4)	.596±.015(4)	.724±.006(4)	3.875
	GLOCAL	.617±.005(4)	.781±.016(2)	.838±.007(4)	.681±.019(3)	.279±.006(4)	.586±.010(3)	.631±.012(2)	.766±.009(3)	3.125
	LSML	.754±.007(2)	.780±.014(3)	.860±.007(2)	.626±.010(4)	.285±.007(2)	**.603±.009(1)**	.630±.012(3)	.777±.009(2)	2.375

TABLE V
PREDICTION RESULTS FOR MULTILABEL LEARNING WITH MISSING 60% LABELS ON FIVE EVALUATION CRITERIA. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD, AND THE NUMBER IN THE BRACKET INDICATES THE RANKING OF THIS ALGORITHM. THE LAST COLUMN SHOWS THE AVERAGE RANKING OF EACH ALGORITHM ON EACH EVALUATION METRIC

		yeast	image	scene	emotions	corel5k	science	education	social	AvgR
$\mathcal{O}\downarrow$	RMFL	**.222±.019(1)**	**.254±.022(1)**	**.181±.019(1)**	.331±.043(2)	**.660±.012(1)**	**.492±.014(1)**	**.461±.016(1)**	**.280±.006(1)**	**1.125**
	FastTag	.557±.019(5)	.438±.028(4)	.271±.036(4)	.684±.080(5)	.839±.024(5)	.681±.018(5)	.710±.016(5)	.460±.020(5)	4.750
	LEML	.342±.015(3)	.471±.031(5)	.395±.021(5)	**.324±.048(1)**	.676±.010(4)	.557±.010(4)	.505±.016(4)	.339±.007(4)	3.750
	GLOCAL	.369±.015(4)	.349±.026(3)	.264±.015(3)	.469±.035(3)	.672±.007(2)	.514±.012(3)	.483±.019(3)	.297±.007(3)	3.000
	LSML	.242±.016(2)	.340±.029(2)	.229±.019(2)	.517±.020(4)	.675±.009(3)	.493±.012(2)	.481±.021(2)	.282±.010(2)	2.375
$\mathcal{H}\downarrow$	RMFL	**.197±.006(1)**	**.154±.006(1)**	**.077±.004(1)**	**.228±.014(1)**	**.009±.000(1)**	**.031±.000(1)**	**.037±.000(1)**	**.020±.000(1)**	**1.125**
	FastTag	.243±.007(2)	.213±.011(3)	.118±.009(3)	.333±.018(5)	.013±.000(5)	.047±.001(4)	.061±.001(5)	.033±.001(5)	4.000
	LEML	.462±.006(5)	.216±.006(4)	.136±.003(4)	.271±.007(2)	**.009±.000(1)**	.056±.001(5)	.052±.001(4)	.028±.001(3)	3.625
	GLOCAL	.303±.004(4)	.248±.004(5)	.178±.001(5)	.311±.007(4)	**.009±.000(1)**	.036±.000(3)	.044±.000(3)	.032±.000(4)	3.750
	LSML	.284±.005(3)	.191±.008(2)	.097±.004(2)	.309±.008(3)	.010±.000(4)	.032±.000(2)	.038±.001(2)	.020±.000(2)	2.500
$\mathcal{R}\downarrow$	RMFL	**.167±.005(1)**	**.140±.010(1)**	**.061±.005(1)**	.203±.016(2)	**.161±.004(1)**	**.111±.007(1)**	**.078±.004(1)**	**.058±.005(1)**	**1.125**
	FastTag	.247±.010(3)	.198±.016(4)	.089±.009(3)	.395±.033(5)	.265±.006(5)	.172±.005(4)	.178±.009(5)	.112±.010(4)	4.125
	LEML	.368±.006(5)	.247±.019(5)	.141±.004(5)	**.191±.012(1)**	.195±.005(3)	.198±.008(5)	.150±.007(4)	.118±.006(5)	4.125
	GLOCAL	.348±.006(4)	.184±.014(2)	.092±.003(4)	.288±.025(3)	.181±.005(2)	.124±.006(2)	.085±.004(2)	.071±.004(2)	2.625
	LSML	.186±.005(2)	.188±.016(3)	.082±.006(2)	.362±.033(4)	.223±.004(4)	.132±.008(3)	.106±.006(3)	.071±.004(3)	3.000
$\mathcal{C}\downarrow$	RMFL	.461±.006(2)	**.171±.012(1)**	**.065±.005(1)**	.329±.018(2)	**.371±.008(1)**	**.158±.007(1)**	**.118±.005(1)**	**.086±.007(1)**	**1.250**
	FastTag	**.415±.015(1)**	.171±.013(2)	.078±.009(2)	.392±.031(3)	.437±.006(3)	.189±.008(4)	.193±.009(4)	.123±.012(4)	2.875
	LEML	.687±.012(5)	.247±.016(5)	.131±.003(5)	**.315±.013(1)**	.447±.009(4)	.260±.009(5)	.206±.007(5)	.159±.008(5)	4.375
	GLOCAL	.625±.009(4)	.202±.016(3)	.091±.003(4)	.405±.030(4)	.402±.009(2)	.169±.006(2)	.124±.005(2)	.102±.006(2)	2.875
	LSML	.489±.009(3)	.210±.016(4)	.083±.005(3)	.453±.029(5)	.513±.008(5)	.187±.008(3)	.158±.006(3)	.106±.005(3)	3.625
$\mathcal{A}\uparrow$	RMFL	**.765±.008(1)**	**.830±.012(1)**	**.891±.010(1)**	.761±.022(2)	**.282±.004(1)**	**.598±.011(1)**	**.645±.013(1)**	**.778±.006(1)**	**1.125**
	FastTag	.544±.012(5)	.732±.018(4)	.840±.020(4)	.521±.041(5)	.169±.012(5)	.445±.010(5)	.432±.014(5)	.632±.020(5)	4.750
	LEML	.609±.007(4)	.706±.018(5)	.760±.010(5)	**.776±.021(1)**	.269±.005(3)	.518±.009(4)	.588±.013(4)	.713±.004(4)	3.750
	GLOCAL	.609±.007(3)	.776±.015(2)	.841±.007(3)	.670±.020(3)	.269±.004(2)	.579±.008(3)	.627±.013(2)	.759±.007(3)	2.625
	LSML	.741±.008(2)	.774±.018(3)	.860±.010(2)	.632±.018(4)	.266±.006(4)	.595±.008(2)	.623±.014(3)	.769±.009(2)	2.750

learning with full labels. In each subfigure, any comparing algorithm whose average rank is within one CD to that of RMFL is connected with RMFL. Otherwise, the algorithm not connected with RMFL is considered to have a significantly different performance from RMFL. As can be seen from Fig. 1, RMFL achieves the best average rank on all the evaluation metrics. It is also worth noting that RMFL significantly outperforms FastTag, GLOCAL, LEML, and JFSC on all the evaluation metrics.

2) Learning With Missing Labels: Since ML^2, LLSF, and JFSC cannot deal with missing labels, we exclude these three approaches in experiments on learning with missing labels. We conduct experiments with the M.L.Ratio ranging in {40%, 60%, 80%}. Tables IV–VI report the prediction results for learning with missing labels when M.L.Ratio is set to 0.4, 0.6, and 0.8, respectively. It can be seen that RMFL is clearly superior to other comparing approaches. The advantage of RMFL over other approaches is that the structural

TABLE VI
PREDICTION RESULTS FOR MULTILABEL LEARNING WITH MISSING 80% LABELS ON FIVE EVALUATION CRITERIA. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD, AND THE NUMBER IN THE BRACKET INDICATES THE RANKING OF THIS ALGORITHM. THE LAST COLUMN SHOWS THE AVERAGE RANKING OF EACH ALGORITHM ON EACH EVALUATION METRIC

		yeast	image	scene	emotions	corel5k	science	education	social	AvgR
$\mathcal{O}\downarrow$	RMFL	**.237±.014(1)**	**.275±.015(1)**	**.192±.021(1)**	.342±.049(2)	**.711±.013(1)**	.523±.017(2)	**.470±.015(1)**	.299±.011(2)	**1.375**
	FastTag	.681±.031(5)	.439±.039(5)	.265±.017(3)	.707±.071(5)	.913±.011(5)	.716±.016(5)	.749±.008(5)	.500±.018(5)	4.750
	LEML	.337±.019(3)	.449±.036(4)	.401±.019(5)	**.333±.035(1)**	.718±.013(3)	.554±.009(4)	.514±.019(4)	.333±.005(4)	3.500
	GLOCAL	.375±.024(4)	.361±.022(2)	.281±.017(4)	.484±.047(3)	.720±.013(4)	.535±.013(3)	.492±.018(3)	.312±.011(3)	3.325
	LSML	.242±.016(2)	.342±.026(3)	.235±.015(2)	.530±.025(4)	.712±.013(2)	**.495±.014(1)**	.485±.017(2)	**.286±.010(1)**	2.125
$\mathcal{H}\downarrow$	RMFL	**.208±.004(1)**	**.188±.009(1)**	**.096±.007(1)**	**.280±.009(1)**	.010±.000(3)	.035±.000(2)	.043±.001(2)	.023±.000(2)	**1.625**
	FastTag	.244±.004(2)	.213±.020(3)	.116±.004(3)	.347±.049(5)	.011±.001(4)	.049±.001(4)	.062±.003(4)	.033±.001(4)	3.625
	LEML	.485±.007(5)	.217±.007(4)	.137±.002(4)	.296±.007(2)	.013±.000(5)	.099±.002(5)	.071±.001(5)	.037±.000(5)	4.375
	GLOCAL	.303±.004(3)	.248±.004(5)	.178±.001(5)	.310±.007(3)	**.009±.000(1)**	.036±.000(3)	.044±.000(3)	.032±.000(3)	3.250
	LSML	.303±.003(4)	**.188±.008(1)**	.098±.003(2)	.312±.007(4)	**.009±.000(1)**	**.032±.000(1)**	**.039±.000(1)**	**.020±.000(1)**	1.875
$\mathcal{R}\downarrow$	RMFL	**.177±.004(1)**	**.146±.005(1)**	**.061±.005(1)**	.214±.016(2)	**.196±.004(1)**	**.106±.007(1)**	**.074±.003(1)**	**.058±.003(1)**	**1.125**
	FastTag	.267±.008(3)	.204±.015(4)	.086±.006(3)	.381±.046(5)	.317±.016(5)	.188±.013(4)	.180±.006(5)	.117±.006(4)	4.125
	LEML	.383±.010(5)	.236±.017(5)	.144±.004(5)	**.195±.017(1)**	.239±.005(3)	.199±.005(5)	.158±.011(4)	.119±.006(5)	4.125
	GLOCAL	.358±.007(4)	.188±.011(3)	.096±.004(4)	.319±.027(3)	.222±.006(2)	.121±.005(2)	.084±.004(2)	.066±.004(2)	2.750
	LSML	.244±.010(2)	.183±.014(2)	.084±.005(2)	.368±.017(4)	.296±.008(4)	.137±.007(3)	.108±.004(3)	.074±.006(3)	2.875
$\mathcal{C}\downarrow$	RMFL	.474±.005(2)	**.174±.009(1)**	**.066±.006(1)**	.341±.016(2)	.468±.011(2)	**.152±.008(1)**	**.110±.005(1)**	**.077±.005(1)**	**1.375**
	FastTag	**.463±.008(1)**	.176±.009(2)	.075±.004(2)	.347±.041(3)	**.377±.015(1)**	.195±.014(3)	.185±.007(4)	.121±.006(4)	2.500
	LEML	.724±.009(5)	.240±.013(5)	.135±.003(5)	**.324±.017(1)**	.541±.009(4)	.263±.007(5)	.215±.012(5)	.160±.007(5)	4.375
	GLOCAL	.657±.008(4)	.206±.012(4)	.094±.002(4)	.429±.027(4)	.505±.011(3)	.167±.005(2)	.119±.004(2)	.094±.003(2)	3.125
	LSML	.597±.012(3)	.206±.015(3)	.085±.004(3)	.455±.014(5)	.646±.010(5)	.198±.008(4)	.162±.005(3)	.111±.007(3)	3.625
$\mathcal{A}\uparrow$	RMFL	**.749±.009(1)**	**.820±.007(1)**	**.887±.011(1)**	.750±.019(2)	**.241±.006(1)**	.580±.010(2)	**.640±.013(1)**	**.771±.008(1)**	**1.250**
	FastTag	.472±.017(5)	.729±.021(4)	.844±.009(3)	.514±.050(5)	.123±.012(5)	.416±.015(5)	.406±.009(5)	.601±.015(5)	4.625
	LEML	.597±.007(4)	.719±.019(5)	.756±.008(5)	**.766±.018(1)**	.231±.005(3)	.517±.007(4)	.581±.014(4)	.716±.008(4)	3.750
	GLOCAL	.602±.010(3)	.769±.012(3)	.832±.008(4)	.650±.024(3)	.231±.006(2)	.562±.009(3)	.621±.013(2)	.753±.008(3)	2.875
	LSML	.704±.013(2)	.777±.017(2)	.857±.008(2)	.624±.010(4)	.221±.005(4)	**.590±.010(1)**	.621±.011(3)	.767±.008(2)	2.500

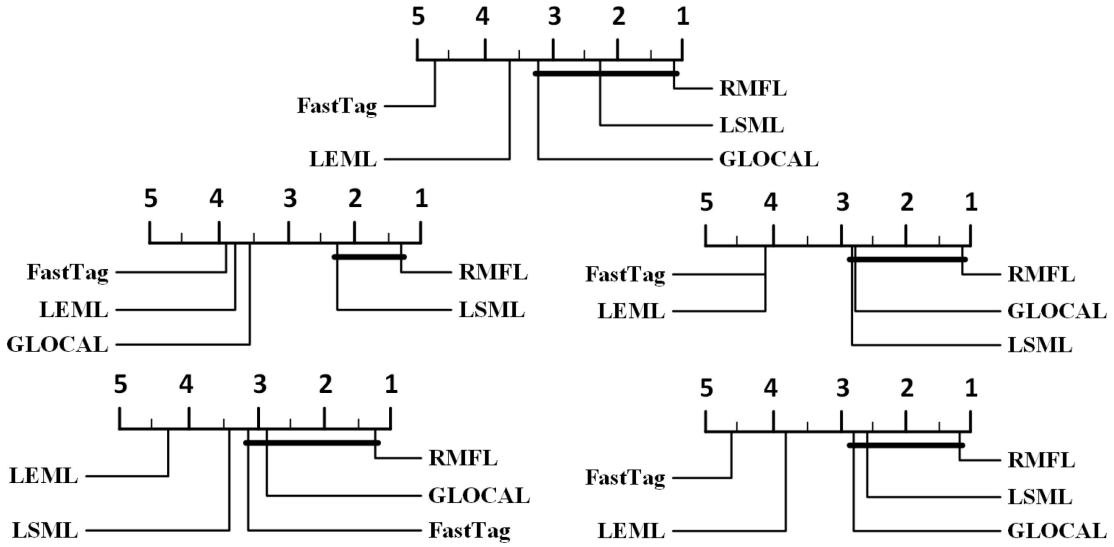


Fig. 2. Statistical comparison of RMFL (the control algorithm) against other comparing algorithms based on the *Nemenyi test* for learning with full labels. Algorithms not connected with RMFL in the CD diagram are considered to have significantly different performance from the control algorithm.

information in the feature space is used to regularize the latent factors of instances. Such well-represented latent factors of instances could recover the ground-truth label matrix with the latent factors of labels while guiding the desired model training simultaneously. In addition, we can observe that predictive performance improves with more observed labels in general. This agrees with the intuition that as more elements in the label matrix are observed, more supervised information can be provided. As shown in Table VII, for learning with missing labels, the null hypothesis that all the comparing algorithms perform equivalently is also rejected on each evaluation metric at the significance level $\alpha = 0.05$. Following the above instructions, for learning with missing labels, we can obtain CD $= 2.15$ with the number of comparing algorithms $A = 5$ and the number of datasets $N = 8$ at the significance level $\alpha = 0.05$. Fig. 2 illustrates the CD diagrams on each evaluation metric for learning with missing labels. As can be seen from Fig. 2, RMFL ranks the first on all the evaluation metrics, and RMFL significantly outperforms FastTag and LEML in most cases. These experimental results clearly demonstrate the effectiveness of the RMFL.

We also conduct additional experiments on four large-scale datasets *pascal07*, *corel16ks1*, *corel16ks2*, and *corel16ks3*. The experimental results are reported in Table VIII. As can be seen from Table VIII, RMFL still outperforms other compared multilabel learning algorithms on these large-scale datasets.

TABLE VII
FRIEDMAN STATISTICS F_F ACCORDING TO EACH EVALUATION METRIC FOR LEARNING WITH MISSING LABELS

	Evaluation metric	F_F	critical value ($\alpha = 0.05$)
Missing labels	One-error	19.21	2.714 ($A = 5, N = 8$)
	Hamming loss	7.27	
	Ranking loss	11.03	
	Coverage	6.79	
	Average precision	13.74	

C. Further Analysis

1) Time Complexity Analysis: The time complexity of RMFL is dominated by matrix multiplication and inversion. In Algorithm 1, the time complexity of learning $\boldsymbol{S}$ and constructing $\boldsymbol{K}$ is $\mathcal{O}(m^2n)$. For each iteration, the time complexity of computing $\nabla \boldsymbol{V}$ and $\nabla \boldsymbol{U}$ is $\mathcal{O}(mlk+l^2k)$, and the time complexity of computing $\nabla \boldsymbol{Z}$ is $\mathcal{O}(mlk+l^2k)$. As has been shown before, it is very flexible for our RMFL algorithm to choose a linear or kernel model. If the linear model is chosen, the time complexities of updating $\boldsymbol{W}$ and $\boldsymbol{b}$ are $\mathcal{O}(mnk + mn^2 + n^3)$ and $\mathcal{O}(mnk)$, respectively; if the kernel model is chosen, the time complexities of updating $\boldsymbol{A}$ and $\boldsymbol{b}$ are $\mathcal{O}(m^3 + m^2k)$ and $\mathcal{O}(m^2k)$, respectively. Since $m > l$, $m > k$, $n > l$, $n > k$ normally hold, the overall time complexity of RMFL is $\min\{\mathcal{O}(Tm^3+m^2n), \mathcal{O}(T(n^3+n^2m)+m^2n)\}$, where T denotes the total number of iterations. Such a conclusion can guide us to choose a linear or kernel model for saving computation time. Specifically, if $n > m$ for a dataset, we can choose a kernel model, and the time complexity will be in the linear order w.r.t. n; otherwise, we can choose a linear model, and the time complexity will be in the quadratic order w.r.t. m. In addition, we will show in the following that RMFL converges very fast (within a few iterations), hence T could be very small.

2) Sensitivity Analysis: There are four regularization parameters $\lambda_1, \lambda_2, \lambda_3$, and λ_4 and two hyperparameters, including the latent dimension k and the number of nearest neighbors K. We provide the sensitivity analysis of these parameters by varying one parameter while keeping others fixed. We first perform sensitivity analysis on the four regularization parameters, and the experimental results are shown in Fig. 3.

It is worth noting that λ_1 and λ_2 control the importance of the latent factors of instances and labels, respectively. It can be seen that the highest performance is usually achieved at some intermediate values of λ_1 and λ_2. This phenomenon clearly validates the importance of learning effective representations of instances and labels in the model training process. Concretely, when λ_1 and λ_2 are too small, the performance of RMFL is not good enough, because the importance of regularization on instances and labels is hardly taken into consideration, and we may not obtain the well-learned representations of instances and labels. While if λ_1 and λ_2 are overly large, the performance of RMFL is also not so good, because we focus too much on learning the representations of instances and labels, and regrettably ignore the importance of training a good model with a good fitting ability. Through the above analysis, the effectiveness of the two regularization terms on instances and labels is clearly demonstrated.

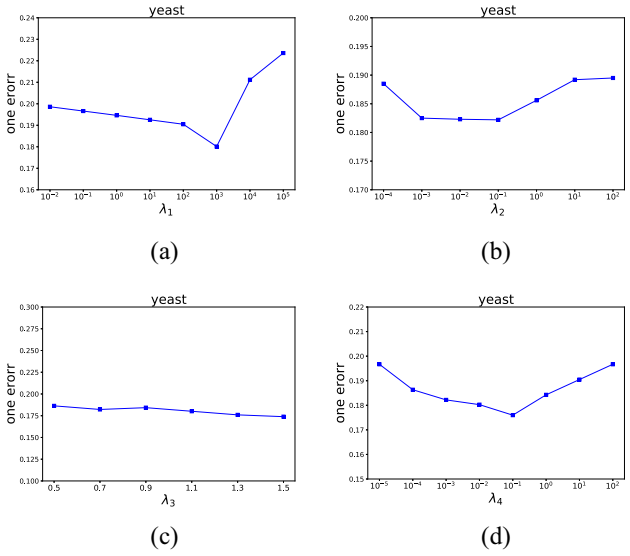


Fig. 3. Sensitivity analysis of the four regularization parameters on *yeast*. (a) Varying λ_1. (b) Varying λ_2. (c) Varying λ_3. (d) Varying λ_4.

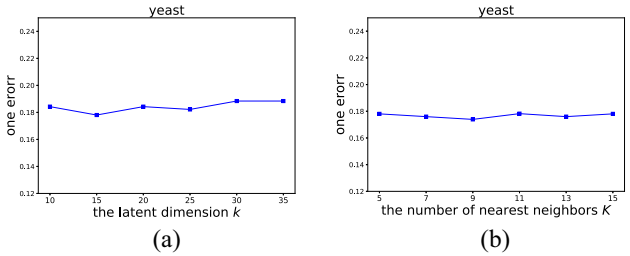


Fig. 4. Sensitivity analysis of the two hyperparameters on *yeast*. (a) Varying k. (b) Varying K.

In addition, we can find that RMFL is relatively insensitive to λ_3. This observation is due to the fact that the ground-truth label matrix can be recovered by both model training and matrix factorization, which means, for recovering the ground-truth label matrix, the two techniques (model training and matrix factorization) play a similar role and there is no obvious advantage of one over the other one. This observation can guide us to easily choose a suitable hyperparameter for λ_3.

Besides, λ_4 controls the model complexity, and we can observe that the performance of RMFL is relatively poor when λ_4 is too small or too large. If λ_4 is too small, RMFL suffers from the overfitting issue. While, if λ_4 is too large, RMFL suffers from the underfitting issue. This observation clearly agrees with our intuition that it is important to balance between overfitting and underfitting.

We also conduct a parameter sensitivity analysis of the two hyperparameters k and K, and the experimental results are shown in Fig. 4. As can be seen from Fig. 4, RMFL is quite robust to k and K. The two hyperparameters are not key parameters of RMFL, while we have to take a value for them (due to KNN and the latent dimension). This observation indicates that we can easily take a suitable value for the two hyperparameters, and alleviate the heavy work of tuning the hyperparameters.

TABLE VIII
PREDICTION RESULTS FOR MULTILABEL LEARNING WITH DIFFERENT RATIOS (40%,60%, AND 80%) OF MISSING LABELS ON FIVE EVALUATION CRITERIA. THE BEST RESULTS ARE HIGHLIGHTED IN BOLD

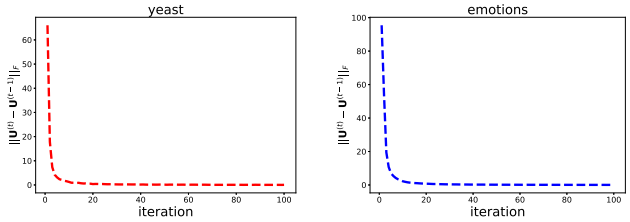
			RMFL	FastTag	LEML	GLOCAL	LSML				RMFL	FastTag	LEML	GLOCAL	LSML
pascal007	$\mathcal{O}\downarrow$	20%	**.484±.014**	.717±.007	.549±.011	.542±.010	.520±.013	crorel16ks1	$\mathcal{O}\downarrow$	20%	.683±.007	.946±.004	.678±.009	.684±.010	**.676±.007**
		40%	**.477±.013**	.558±.007	.541±.013	.542±.009	.520±.010			40%	**.660±.007**	.731±.007	.669±.008	.661±.008	.663±.008
		60%	**.476±.011**	.576±.008	.534±.011	.541±.009	.523±.010			60%	**.649±.006**	.685±.007	.654±.010	.656±.009	.653±.009
	$\mathcal{H}\downarrow$	20%	.074±.001	.116±.006	.072±.001	.074±.001	**.070±.001**		$\mathcal{H}\downarrow$	20%	**.018±.000**	.027±.001	.019±.000	.019±.000	.019±.000
		40%	**.065±.001**	.100±.006	.072±.001	.074±.001	.070±.001			40%	**.018±.000**	.028±.001	.019±.000	.019±.000	.019±.000
		60%	**.064±.001**	.102±.003	.069±.001	.074±.001	.068±.001			60%	**.018±.000**	.030±.001	.019±.000	.019±.000	.019±.000
	$\mathcal{R}\downarrow$	20%	**.143±.004**	.240±.003	.192±.004	.174±.005	.165±.005		$\mathcal{R}\downarrow$	20%	.206±.003	.341±.002	.208±.005	**.194±.002**	.227±.005
		40%	**.141±.003**	.174±.003	.188±.006	.169±.006	.162±.005			40%	**.172±.003**	.227±.003	.184±.004	.173±.003	.195±.003
		60%	**.141±.003**	.170±.003	.181±.005	.167±.005	.161±.005			60%	**.168±.004**	.199±.003	.174±.003	.170±.004	.181±.004
	$\mathcal{C}\downarrow$	20%	**.195±.006**	.294±.003	.253±.006	.232±.006	.222±.006		$\mathcal{C}\downarrow$	20%	.402±.011	.508±.004	.411±.007	**.383±.004**	.452±.008
		40%	**.193±.005**	.224±.004	.247±.007	.225±.008	.218±.006			40%	**.334±.006**	.424±.006	.362±.008	.335±.007	.389±.007
		60%	**.191±.005**	.220±.004	.236±.005	.219±.006	.213±.006			60%	.333±.006	.385±.007	.342±.005	**.326±.006**	.362±.007
	$\mathcal{A}\uparrow$	20%	**.599±.008**	.419±.004	.534±.007	.534±.006	.559±.007		$\mathcal{A}\uparrow$	20%	**.306±.004**	.089±.002	.303±.004	.298±.004	.300±.005
		40%	**.605±.008**	.540±.004	.539±.010	.540±.008	.561±.007			40%	**.325±.004**	.252±.002	.321±.004	.322±.003	.320±.003
		60%	**.605±.006**	.526±.004	.545±.007	.545±.007	.560±.007			60%	**.335±.004**	.295±.004	.333±.005	.327±.004	.333±.005
crorel16ks2	$\mathcal{O}\downarrow$	20%	**.673±.012**	.945±.006	.678±.007	.674±.010	.678±.010	crorel16ks3	$\mathcal{O}\downarrow$	20%	**.670±.010**	.952±.004	.677±.011	.676±.005	.675±.010
		40%	**.648±.010**	.712±.012	.660±.009	.650±.009	.654±.007			40%	**.656±.009**	.720±.008	.664±.007	.657±.009	.659±.006
		60%	**.637±.008**	.668±.007	.642±.007	.643±.008	.639±.006			60%	**.643±.004**	.669±.007	.647±.008	.649±.009	.644±.010
	$\mathcal{H}\downarrow$	20%	**.018±.000**	.026±.001	**.018±.000**	**.018±.000**	**.018±.000**		$\mathcal{H}\downarrow$	20%	**.018±.000**	.027±.001	**.018±.000**	**.018±.000**	**.018±.000**
		40%	**.017±.000**	.026±.001	.018±.000	.018±.001	.018±.000			40%	**.018±.000**	.030±.001	**.018±.000**	**.018±.001**	**.018±.000**
		60%	**.017±.000**	.028±.001	.018±.000	.018±.001	**.017±.000**			60%	**.017±.000**	.029±.001	.018±.000	.018±.000	.018±.000
	$\mathcal{R}\downarrow$	20%	**.187±.005**	.342±.003	.205±.004	.190±.003	.228±.005		$\mathcal{R}\downarrow$	20%	**.188±.004**	.342±.002	.201±.003	.189±.004	.223±.003
		40%	**.168±.003**	.225±.003	.180±.003	.171±.006	.193±.004			40%	**.168±.003**	.225±.002	.180±.003	**.168±.006**	.191±.003
		60%	**.156±.001**	.196±.003	.171±.004	.160±.004	.180±.003			60%	**.157±.001**	.197±.001	.170±.002	.169±.006	.178±.002
	$\mathcal{C}\downarrow$	20%	**.377±.012**	.510±.005	.407±.006	.379±.005	.455±.008		$\mathcal{C}\downarrow$	20%	.397±.008	.506±.003	.401±.005	**.378±.006**	.447±.006
		40%	.337±.004	.423±.005	.357±.006	**.333±.011**	.389±.007			40%	**.326±.004**	.420±.003	.357±.006	.328±.010	.384±.005
		60%	.312±.003	.378±.005	.337±.006	**.305±.006**	.361±.006			60%	**.323±.004**	.380±.003	.337±.003	.327±.011	.358±.005
	$\mathcal{A}\uparrow$	20%	**.303±.008**	.084±.002	.298±.004	.295±.007	.294±.005		$\mathcal{A}\uparrow$	20%	.299±.003	.082±.002	**.304±.003**	.298±.005	.300±.004
		40%	**.323±.007**	.257±.005	.321±.005	.320±.007	.320±.005			40%	**.329**±.007	.254±.003	.321±.004	.322±.005	.322±.004
		60%	**.339±.004**	.299±.004	.333±.005	.329±.005	.333±.005			60%	**.340±.003**	.301±.002	.334±.003	.325±.005	.335±.004

TABLE IX
ABLATION STUDY OF RMFL ON *yeast* AND *image*

		RMFL	RMFL $\lambda_1 = 0$	RMFL $\lambda_2 = 0$	RMFL Linear
yeast	$\mathcal{O}\downarrow$	**0.186±0.009**	0.201±0.010	0.204±0.009	0.223±0.013
	$\mathcal{H}\downarrow$	**0.182±0.006**	0.184±0.006	0.185±0.006	0.200±0.005
	$\mathcal{R}\downarrow$	**0.153±0.005**	0.155±0.005	0.158±0.005	0.172±0.005
	$\mathcal{C}\downarrow$	**0.432±0.007**	0.438±0.007	0.441±0.007	0.463±0.006
	$\mathcal{A}\uparrow$	**0.787±0.007**	0.782±0.007	0.780±0.007	0.763±0.008
image	$\mathcal{O}\downarrow$	**0.235±0.020**	0.238±0.020	0.240±0.021	0.331±0.019
	$\mathcal{H}\downarrow$	**0.146±0.008**	0.150±0.008	0.148±0.008	0.186±0.010
	$\mathcal{R}\downarrow$	**0.131±0.009**	0.134±0.009	0.138±0.011	0.179±0.013
	$\mathcal{C}\downarrow$	**0.166±0.012**	0.169±0.012	0.173±0.013	0.198±0.014
	$\mathcal{A}\uparrow$	**0.842±0.010**	0.839±0.010	0.836±0.013	0.786±0.011

3) Ablation Study: Here, we provide an ablation study that demonstrates the effectiveness of respective parts in our RMFL algorithm. We compare RFML with three weakened versions, including: 1) removing the regularization on instances ($\lambda_1 = 0$); 2) removing the regularization on labels ($\lambda_2 = 0$); and 3) using the linear model instead of the kernel model. The experimental results of the ablation study are reported in Table IX. As shown in Table IX, the original RMFL algorithm always achieves the best performance. Besides, RMFL with $\lambda_1 = 0$ achieves similar performance as RMFL with $\lambda_2 = 0$, which demonstrates that the regularization on instances is as important as the regularization on labels. In addition, RMFL with the kernel model clearly outperforms RMFL with the linear model.

4) Convergence Analysis: We empirically study the convergence of RMFL. Fig. 5 shows the difference of the variable $\boldsymbol{U}$ between two successive iterations on *yeast* and *emotions*. As can be seen, RMFL converges within a few iterations. Such convergence trends can be also observed on other datasets.



(a) (b)

Fig. 5. Convergence analysis of RMFL on *yeast* and *emotions*. (a) Convergence trend on *yeast*. (b) Convergence trend on *emotions*.

VI. CONCLUSION

In this article, we proposed a regularized matrix factorization-based method to solve the problem of MLMLs. By decomposing the observed label matrix into two matrices, the latent factors of instances and labels can be obtained, which were regularized by the structural information in feature space and the latent label correlations, respectively. In this way, the ground-truth label matrix was recovered while the desired model was trained simultaneously. Experiments on both full-label data and missing-label data demonstrated the effectiveness of RMFL.

In the future, we will explore if there exist better ways to exploit the feature space for recovering the label space. Besides, since our RMFL method could converge to a local minimum, it would be very interesting to show that the obtained local minima by the alternating optimization method would be provable to achieve satisfying performance in future work. In addition, it is worth noting that both the linear model and the kernel model in our RMFL method are shallow models. Therefore, we will also explore the deep neural networks to further improve the practical performance.

ACKNOWLEDGMENT

Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

REFERENCES

[1] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 8, pp. 1819–1837, Aug. 2014.

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

[3] J. Du and C.-M. Vong, "Robust online multilabel learning under dynamic changes in data distribution with labels," *IEEE Trans. Cybern.*, vol. 50, no. 1, pp. 374–385, Jan. 2020.

[4] C. Zhang, Z. Yu, H. Fu, P. Zhu, L. Chen, and Q. Hu, "Hybrid noise-oriented multilabel learning," *IEEE Trans. Cybern.*, vol. 50, no. 6, pp. 2837–2850, Jun. 2020.

[5] M. Xu, Y.-F. Li, and Z.-H. Zhou, "Robust multi-label learning with PRO loss," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 8, pp. 1610–1624, Aug. 2020.

[6] G. Tsoumakas and I. Katakis, "Multi-label classification: An overview," *Int. J. Data Warehousing Min.*, vol. 3, no. 3, pp. 1–13, 2007.

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

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

[9] M. P. Kumar, B. Packer, and D. Koller, "Self-paced learning for latent variable models," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2010, pp. 1189–1197.

[10] Y.-K. Li, M.-L. Zhang, and X. Geng, "Leveraging implicit relative labeling-importance information for effective multi-label learning," in *Proc. IEEE Int. Conf. Data Min.*, Atlantic City, NJ, USA, 2015, pp. 251–260.

[11] H. Yang, J. T. Zhou, Y. Zhang, B.-B. Gao, J. Wu, and J. Cai, "Exploit bounding box annotations for multi-label object recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Las Vegas, NV, USA, 2016, pp. 280–288.

[12] L. Tang, S. Rajan, and V. K. Narayanan, "Large scale multi-label classification via metalabeler," in *Proc. 18th Int. Conf. World Wide Web*, 2009, pp. 211–220.

[13] Z. Barutcuoglu, R. E. Schapire, and O. G. Troyanskaya, "Hierarchical multi-label prediction of gene function," *Bioinformatics*, vol. 22, no. 7, pp. 830–836, 2006.

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

[15] Y.-Y. Sun, Y. Zhang, and Z.-H. Zhou, "Multi-label learning with weak label," in *Proc. AAAI Conf. Artif. Intell.*, 2010, pp. 593–598.

[16] L. Xu, Z. Wang, Z. Shen, Y. Wang, and E. Chen, "Learning low-rank label correlations for multi-label classification with missing labels," in *Proc. IEEE Int. Conf. Data Min.*, Shenzhen, China, 2014, pp. 1067–1072.

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

[18] F. Zhao and Y. Guo, "Semi-supervised multi-label learning with incomplete labels," in *Proc. 24th Int. Joint Conf. Artif. Intell.*, 2015, pp. 4062–4068.

[19] L. Feng, B. An, and S. He, "Collaboration based multi-label learning," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 3550–3557.

[20] M. Xu, R. Jin, and Z.-H. Zhou, "Speedup matrix completion with side information: Application to multi-label learning," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2013, pp. 2301–2309.

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

[22] Y.-N. Chen and H.-T. Lin, "Feature-aware label space dimension reduction for multi-label classification," in *Advances in Neural Information Processing Systems*. Red Hook, NY, USA: Curran, 2012, pp. 1529–1537.

[23] J. Wicker, B. Pfahringer, and S. Kramer, "Multi-label classification using Boolean matrix decomposition," in *Proc. 27th Annu. ACM Symp. Appl. Comput.*, 2012, pp. 179–186.

[24] Y. Zhang and J. Schneider, "Maximum margin output coding," in *Proc. Int. Conf. Mach. Learn.*, 2012, pp. 1575–1582.

[25] Z. Lin, G. Ding, M. Hu, and J. Wang, "Multi-label classification via feature-aware implicit label space encoding," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 325–333.

[26] W. Liu and I. W. Tsang, "Large margin metric learning for multi-label prediction," in *Proc. AAAI Conf. Artif. Intell.*, 2015, pp. 2800–2806.

[27] X. Shen, W. Liu, I. W. Tsang, Q.-S. Sun, and Y.-S. Ong, "Multilabel prediction via cross-view search," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 29, no. 9, pp. 4324–4338, Sep. 2018.

[28] K.-H. Huang and H.-T. Lin, "Cost-sensitive label embedding for multi-label classification," *Mach. Learn.*, vol. 106, nos. 9–10, pp. 1725–1746, 2017.

[29] F. Tai and H.-T. Lin, "Multilabel classification with principal label space transformation," *Neural Comput.*, vol. 24, no. 9, pp. 2508–2542, Sep. 2012.

[30] P. Hou, X. Geng, and M.-L. Zhang, "Multi-label manifold learning," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 1680–1686.

[31] S. He, L. Feng, and L. Li, "Estimating latent relative labeling importances for multi-label learning," in *Proc. IEEE Int. Conf. Data Min.*, Singapore, 2018, pp. 1013–1018.

[32] Q.-W. Zhang, Y. Zhong, and M.-L. Zhang, "Feature-induced labeling information enrichment for multi-label learning," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 4446–4453.

[33] S. S. Bucak, R. Jin, and A. K. Jain, "Multi-label learning with incomplete class assignments," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Providence, RI, USA, 2011, pp. 2801–2808.

[34] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, no. 5500, pp. 2323–2326, 2000.

[35] N. Boumal, B. Mishra, P.-A. Absil, and R. Sepulchre, "Manopt, a MATLAB toolbox for optimization on manifolds," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1455–1459, 2014.

[36] J. Huang *et al.*, "Improving multi-label classification with missing labels by learning label-specific features," *Inf. Sci.*, vol. 492, pp. 124–146, Aug. 2019.

[37] J. Huang, G. Li, Q. Huang, and X. Wu, "Joint feature selection and classification for multilabel learning," *IEEE Trans. Cybern.*, vol. 48, no. 3, pp. 876–889, Mar. 2018.

[38] L. Feng and B. An, "Partial label learning with self-guided retraining," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 3542–3549.

[39] L. Feng and B. An, "Partial label learning by semantic difference maximization," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 2294–2300.

[40] B. Schölkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA, USA: MIT Press, 2002.

[41] M. Chen, A. Zheng, and K. Weinberger, "Fast image tagging," in *Proc. Int. Conf. Mach. Learn.*, 2013, pp. 1274–1282.

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

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

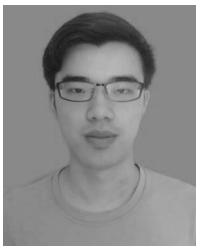
Lei Feng received the B.E. degree in computer science from Southwest University, Chongqing, China, in 2017. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, Nanyang Technological University, Singapore.

He has published ten papers on top conferences, such as ICML, AAAI, IJCAI, CVPR, and ICDM. His main research interests include weakly supervised learning, statistical learning theory, domain adaptation, and data mining.

Mr. Feng has also served as a Conference Program Committee Member for NeurIPS, ICLR, and AAAI.

Jun Huang received the M.S. degree in computer science from the Anhui University of Technology, Ma'anshan, China, in 2011, and the Ph.D. degree in computer science from the University of Chinese Academy of Sciences, Beijing, China, in 2017.

He is currently an Assistant Professor with the School of Computer Science and Technology, Anhui University of Technology. His research interests include machine learning and data mining.



Senlin Shu received the B.E. degree in energy and power engineering from Sichuan University, Chengdu, China, in 2017. He is currently pursuing the master's degree with the College of Computer and Information Science, Southwest University, Chongqing, China.

His main research interests include partial-label learning and domain adaptation.

Bo An received the Ph.D. degree in computer science from the University of Massachusetts, Amherst, MA, USA, in 2010.

He is a President's Council Chair Associate Professor of computer science and engineering with Nanyang Technological University, Singapore. He has published over 100 referred papers at *Autonomous Agents and Multiagent Systems*, IJCAI, AAAI, ICAPS, KDD, UAI, EC, WWW, ICLR, NeurIPS, ICML, the *Journal of Autonomous Agents and Multiagent Systems*, *Artificial Intelligence*, and ACM/IEEE TRANSACTIONS. His current research interests include artificial intelligence, multiagent systems, computational game theory, reinforcement learning, and optimization.

Dr. An was the recipient of the 2010 IFAAMAS Victor Lesser Distinguished Dissertation Award, the Operational Excellence Award from the Commander, the First Coast Guard District of the United States, the 2012 INFORMS Daniel H. Wagner Prize for Excellence in Operations Research Practice, and the 2018 Nanyang Research Award (Young Investigator). His publications won the Best Innovative Application Paper Award at AAMAS'12 and the Innovative Application Award at IAAI'16. He was invited to give Early Career Spotlight talk at IJCAI'17. He led the team HogRider which won the 2017 Microsoft Collaborative AI Challenge. He was named to IEEE Intelligent Systems' "AI's 10 to Watch" list for 2018. He is the PC Co-Chair of AAMAS'20. He is a member of the editorial board of the *Journal of Artificial Intelligence Research* and an Associate Editor of the *Journal of Autonomous Agents and Multiagent Systems*, IEEE INTELLIGENT SYSTEMS, and the *ACM Transactions on Intelligent Systems and Technology*. He was elected to the board of directors of IFAAMAS and a Senior Member of AAAI.