Semi-Supervised Partial Multi-Label Learning

Ming-Kun Xie and Sheng-Jun Huang
College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics
MIIT Key Laboratory of Pattern Analysis and Machine Intelligence
Collaborative Innovation Center of Novel Software Technology and Industrialization, Nanjing, China
{mkxie,huangsj}@nuaa.edu.cn

***Abstract*—Partial multi-label learning (PML) deals with problems where each instance is associated with a candidate label set, which contains multiple relevant labels and some noisy labels. In many real-world scenarios, it is impractical to annotate all examples for a huge-size dataset. Instead, a more common case is that only a small set of the data are annotated with partial labels, while most data are unlabeled. In this paper, we formalize such problems as a new learning framework called Semi-Supervised Partial Multi-label Learning (SSPML). To solve the SSPML problem, a latent label variable is introduced for each example as the low-dimensional embedding of the feature space. On one hand, label variables are recovered by encouraging consistent similarity measurement between the feature space and the label space; on the other hand, the similarities are adaptively updated based on the feedback from the label space. Meanwhile, the multi-label classifier is jointly trained under the supervision of label variables. Extensive experiments on multiple datasets from various real-world tasks validate the effectiveness of the proposed approach.**

I. Introduction

In many real-world classification applications, an instance could be assigned with multiple class labels simultaneously [1]. For example, a piece of music can be categorized into different genres [2]; a page of website may be associated with multiple topics [3], and an image can be annotated with several tags [4]. The task of multi-label learning is to train a classification model that can predict all the relevant labels for unseen instances. In previous studies on multi-label learning, a common assumption is that each training instance has been precisely annotated with all of its relevant labels. However, in many real-world scenarios, one can only get access to a candidate label set for each training instance, which contains multiple relevant labels and some other noisy labels. For example, in crowdsourcing environments, unreliable annotators may assign an image with multiple candidate labels among which only some of them are accurate ones. In order to handle such problems, the partial multi-label learning (PML) framework has been firstly formalized by [5], and in consequence, several advanced PML methods has been recently proposed [6]–[18].

Typical partial multi-label learning methods assume that the candidate label set is available for all training instances. Unfortunately, in many real-world tasks such as video character classification [12] and gene function prediction [7], this assumption hardly holds since it is difficult to annotate all examples in a huge-size dataset. While it is difficult to train effective models based only on the small set of partial-labeled examples, it is rather important to exploit information from unlabeled instances, which are usually easy to collect.

We formalize the learning problem as a new framework called semi-supervised partial multi-label learning (SSPML). More specifically, SSPML attempts to learn a classification model from partially labeled and unlabeled training examples simultaneously, where each instance is either associated with a candidate label set or even without any supervised information. Note that neither PML nor semi-supervised multi-label learning (SSMLL) [19], [20] can be directly applied to solve this problem. On one hand, PML methods fail to utilize the enormous unlabeled data which may be useful for model training; on the other hand, SSMLL always assume that each training instance is associated with ground-truth labels, which is not available in our situation. Therefore, SSPML is a novel learning framework with significant differences with existing settings.

To solve SSPML problems, in this paper, a novel method is proposed to identify ground-truth labels from candidate sets of partially labeled data and meanwhile exploit the manifold structure of unlabeled data. In the proposed SSPML method, a basic assumption is that the ground-truth labels can be regarded as a low-dimensional embedding of the high dimensional feature space. Then by encouraging consistent similarity measurement between the feature space and the label space, a latent label variable is learned for each training instance. Further, the similarities among training examples are optimized by exploiting the information from both feature space and label space. Instead of separating the learning process into two stages, the proposed method performs label variable recovery and multi-label classifier training in a pipeline to facilitate the model to be more robust and generalize well. Extensive experiments on multiple datasets from various real-world tasks demonstrate the effectiveness of the proposed SSPML method. Figure I illustrates the difference between SSPML framework proposed in this paper and its related multi-label learning frameworks.

The rest of this paper is organized as follows: Section 2 reviews some related works; Section 3 introduces our proposed SSPML approach; experimental results are reported in Section 4, followed by the conclusion in Section 5.

II. Related Works

The proposed semi-supervised partial multi-label learning framework is related to two popular learning frameworks:

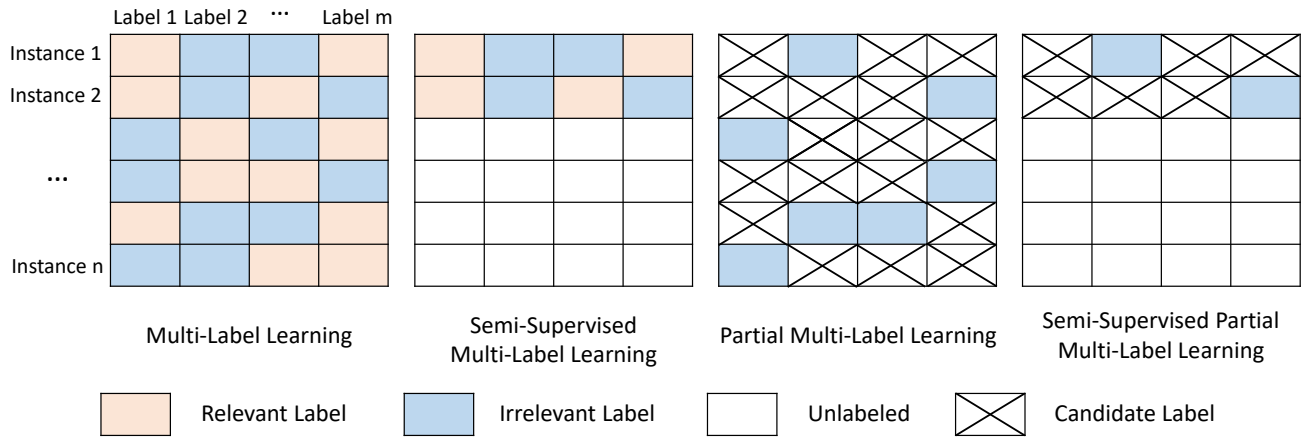


Fig. 1. Comparisons among four multi-label learning frameworks.

semi-supervised multi-label learning and partial multi-label learning.

Exploiting unlabeled data under the multi-label learning setting has attracted many research interests. A few attempts have made to tackle inductive semi-supervised multi-label learning [19]–[21]. In [19], authors propose to learn a subspace representation by utilizing both labeled and unlabeled data, while a classifier is trained simultaneously via large margin criterion on labeled data. In [20], authors try to utilize label correlation in labeled data and maximum-margin regularization over unlabeled data to optimize a group of linear predictors for inductive multi-label classification. A Bayesian semi-supervised multi-label learning (BSSML) is proposed by combining linear dimensionality reduction with linear binary classification under low-density assumption [22]. In [23], a co-training based semi-supervised multi-label learning method is proposed to train two classifiers by dichotomizing the feature space with diversity maximization, and then pairwise ranking predictions on unlabeled data is iteratively communicated for model refinement. As an advanced version, MLCT [24] leverages information concerning the co-occurrence of pairwise labels to address the class-imbalance challenge. DRML [25] is proposed to solve semi-supervised multi-label learning problems by jointly exploring feature distribution and label relation simultaneously. In [26], authors solve semi-supervised multi-label learning problems with missing labels.

To handle the partial-labeled data, one of the most straightforward methods is to train a multi-label classifier by treating all the candidate labels as accurate. Unfortunately, such methods ignore the noisy labels in candidate sets, and thus may result in degenerated performance. In consequence, some techniques are specially designed for solving PML problems recently. Among them, PML-*lc* and PML-*fp* [5] are the firstly proposed two effective methods to solve PML problems by introducing a confidence value for each candidate label. In [7], authors propose to achieve disambiguation by utilizing low-rank matrix approximation and latent dependencies between labels and features. The decomposition scheme is employed to transform the observed noisy label matrix into a low-rank ground-truth label matrix and a sparse noisy label matrix, in which the ground-truth label matrix is used to train a multi-label classifier [10]. PARTICLE [8] identifies the credible labels with high labeling confidences by employing an iterative label propagation procedure.Then, the credible labels are employed to instantiate two PML methods PAR-VLS and PAR-MAP via pairwise label ranking. DRAMA [11] trains a gradient boosting model to fit the label confidence learned from manifold structure in the feature space. PML-NI [6] trains noisy label identifier and multi-label classifier jointly under supervision of the observed label matrix. In [27], authors extend partial multi-label learning into multi-view settings. Despite the advances that these methods have achieved, a potential limitation is that they do not consider solving PML problems with unlabeled data, which cannot apply directly to the problem concerned in this paper.

III. The Proposed Method

In semi-supervised partial multi-label learning, we consider a set of n_p partially labeled instances $\mathcal{D}_p = \{(\boldsymbol{x}_i, \boldsymbol{y}_i)\}_{i=1}^{n_p}$ and a large set of n_u unlabeled instances $\mathcal{D}_u = \{\boldsymbol{x}_i\}_{i=n_l+1}^{n=n_p+n_u}$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ is the feature vector and $\boldsymbol{y}_i \in \{1, 0\}^q$ is the label vector with q class labels for the i-th instance. By arranging feature vectors and label vectors of n training instances, we obtain the feature matrix $\mathbf{X} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_n] \in \mathbb{R}^{d\times n}$ and label matrix $\mathbf{Y} = [\boldsymbol{y}_1, ..., \boldsymbol{y}_n] \in \mathbb{R}^{q\times n}$, where $y_{ji} = 1$ if the j-th label is a candidate label for i-th instance; otherwise, $y_{ji} = 0$. Note that $y_{ji} = -1$, $\forall 1 \leq j \leq q$, $n_p + 1 \leq i \leq n$, which indicates labels are unknown for unlabeled examples.

To deal with SSPML problems, one intuitive baseline method is to disambiguate the candidate label set of the partial-labeled training examples, i.e., identify all the relevant labels from the candidate label set. Then, the original problem is transformed into a semi-supervised multi-label learning (SSMLL) problem, which can be effectively solved by off-

the-shelf SSMLL methods. Unfortunately, such a strategy separates the learning process into two stage, i.e., the candidate label disambiguation and unlabeled data exploitation, which may make the two components inconsistent, and subsequently hurt the generalization performance. Unlike the two-stage strategy, the proposed SSPML method implements candidate label disambiguation and unlabeled data exploitation in a joint framework. Specifically, we regard the ground-truth labels as the latent variables, which form a low-dimensional embedding space. Then by enforcing consistent similarity structure among both partial-labeled and unlabeled examples between the embedding space and feature space, the ground-truth labels are recovered from the observed partial labels. Finally, we incorporate the multi-label classifier training and the label recovery into an unified optimization framework. In the following contents of this section, we will introduce the three components of similarity estimation, the ground-truth label recovery and classifier training, respectively, and finally present the optimization steps.

A. Similarity Estimation

Firstly, we introduce the similarity estimation by implementing the feature sparse reconstruction [7]. Let $\mathbf{S} = [s_{ij}]_{n\times n}$ denote the similarity measurement matrix among training examples, where s_{ij} reflects the similarity degree between the i-th instance and the j-th instance. Guided by the assumption that relationship between one instance and all the other instances can be determined by the contribution of other instances to the reconstruction of this instance, the similarity matrix $\mathbf{S}$ is instantiated by implementing sparse reconstruction between this instance and all the other instances. Let $\mathbf{X}_{-i} = [\boldsymbol{x}_1, ..., \boldsymbol{x}_{i-1}, \boldsymbol{x}_{i+1}, ..., \boldsymbol{x}_n]$ denote the $d \times (n-1)$ feature matrix including all the instances other than $\boldsymbol{x}_i$ and $\boldsymbol{s}_i = [s_{1i}, ..., s_{i-1,i}, s_{i+1,i}, ..., s_{n,i}]^\top$ denote the $(d-1)$-dimensional similarity measurement vector. By implementing sparse reconstruction, the similarity measurement vector $\boldsymbol{s}_i$ can be learned by solving the following optimization problem:

$$\min_{\boldsymbol{s}_i} \frac{1}{2} \left\| \mathbf{X}_{-i}\boldsymbol{s}_i - \boldsymbol{x}_i \right\|_2^2 + \alpha \left\| \boldsymbol{s}_i \right\|_1 \tag{1}$$

Here, the first term controls the reconstruction error to obtain a precise similarity measurement among training instances via ℓ_2 norm, and the second term controls the sparsity of reconstruction via ℓ_1 norm. The relative importance is balanced by the trade-off parameter α.

B. Ground-truth Label Recovery

As previously discussed, the ground-truth label space can be regarded as a low-dimensional embedding space of the high-dimensional feature space. To capture the labeling confidence of each candidate label or unknown label, we introduce a label variable vector $\boldsymbol{z}_i$ for each training instance $\boldsymbol{x}_i$. In other words, $\boldsymbol{z}_{ji}$ measures how likely the j-th label is a ground-truth label of $\boldsymbol{x}_i$. By arranging the label variables of n training examples, we obtain the label variable matrix $\mathbf{Z} = [\boldsymbol{z}_i, ..., \boldsymbol{z}_n] \in [0,1]^{q\times n}$. As the ground-truth labels are noise-free, we can expect that the similarity structure in the feature space still maintains in the embedding space. In consequence, the ground-truth labels of each training instance can be recovered by reconstruction of label variable vectors in low-dimensional embedding space as that of feature vectors have done in the high-dimensional feature space. More specifically, we can recover the label variable matrix by minimizing reconstruction error in the latent space:

$$\mathcal{L}(\boldsymbol{z}_i) = \left\| \mathbf{Z}_{-i}\boldsymbol{s}_i - \boldsymbol{z}_i \right\|_2^2 .$$

However, the objective function ignores the label correlation, which turns out to be an indispensable element in multi-label learning [28], [29]. In order to make full use of label correlation for enhancing label variable reconstruction, for each label variable vector, the objective can be re-written as following:

$$\min_{\boldsymbol{z}_i} \frac{1}{2} \left\| \mathbf{L}\mathbf{Z}_{-i}\boldsymbol{s}_i - \boldsymbol{z}_i \right\|_2^2 . \tag{2}$$

Here, $\mathbf{L} \in \mathbb{R}^{q\times q}$ is the label correlation matrix, where l_{ij} indicate the label correlation between i-th label and j-th label. The matrix can be obtained in various ways, such as co-occurrence matrix [4], [30].

C. MLL Classifier Training

To recover the label variables, one straightforward choice is to solve optimization problem (1) and eq.(2) sequentially, which recovers label variables with a fixed similarity measurement. Unfortunately, such a method suffers from perturbation of noisy data, such as outliers, which may lead to degraded performance. In order to disambiguate correctly for partial-labeled data and obtain a more accurate estimation for unlabeled data, we consider learning similarity weights in an adaptive way. By solving the optimization problem (1) and (2) jointly, the similarities are not only determined by the manifold structure of the feature space, but also guided by the feedback from the label space.

Furthermore, instead of training the classifier independently, we perform these three procedures, i.e., similarity estimation, label variable recovery and classifier training, in a pipeline, which makes them work consistently and benefit from each other. By introducing a multi-label classifier $\mathbf{W} \in \mathbb{R}^{q\times d}$ and $\mathbf{b} \in \mathbb{R}^q$, the objective function of the unified framework which consists of these three parts can be formulated as follows:

$$\begin{aligned} \min_{\mathbf{S},\mathbf{Z},\mathbf{W},\mathbf{b}} & \tfrac{\lambda}{2} \left\| \mathbf{X}\mathbf{S} - \mathbf{X} \right\|_{\mathrm{F}}^2 + \tfrac{\beta}{2} \left\| \mathbf{L}\mathbf{Z}\mathbf{S} - \mathbf{Z} \right\|_{\mathrm{F}}^2 + \alpha \left\| \mathbf{S} \right\|_1 \\ & + \tfrac{\gamma}{2} \left\| \mathbf{Z} - (\mathbf{W}\mathbf{X} + \mathbf{b}\mathbf{1}_n^\top) \right\|_{\mathrm{F}}^2 + \tfrac{\mu}{2} \left\| \mathbf{W} \right\|_{\mathrm{F}}^2 \\ & + \tfrac{1}{2} \left\| \mathbf{J} \circ (\mathbf{Z} - \mathbf{Y}) \right\|_{\mathrm{F}}^2 \\ & s.t. \quad s_{ii} = 0, \forall 1 \le i \le n \end{aligned} \tag{3}$$

Here, $\mathbf{J}$ is a indicator matrix, where $J_{ji} = 1$ if j-th label is not candidate to i-th instance, $\forall 1 \le i \le n_p$; otherwise, $J_{ji} = 0$. Note that $J_{ji} = 0$, $\forall 1 \le j \le q$, $n_p + 1 \le i \le n$.

D. Alternating Optimization

1) Updating $\mathbf{S}$*:* With $\mathbf{Z}$, $\mathbf{W}$ and $\mathbf{b}$ fixed, for the similarity measurement matrix $\mathbf{S}$, the optimization problem (3) can be reformulated as following:

$$\min_{\mathbf{S},\mathbf{Z},\mathbf{W},\mathbf{b}} \tfrac{\lambda}{2}\left\|\mathbf{XS}-\mathbf{X}\right\|_{\mathrm{F}}^2+\tfrac{\beta}{2}\left\|\mathbf{LZS}-\mathbf{Z}\right\|_{\mathrm{F}}^2+\alpha\left\|\mathbf{S}\right\|_1 \quad s.t. \quad s_{ii}=0,\forall 1\le i\le n$$

To solve the problem, we employ the popular Alternating Direction Method of Multiplier (ADMM) [31], which reformulate the above optimization problem into the following equivalent form:

$$\min_{\mathbf{S},\mathbf{Z},\mathbf{W},\mathbf{b}} \tfrac{\lambda}{2}\left\|\mathbf{XS}-\mathbf{X}\right\|_{\mathrm{F}}^2+\tfrac{\beta}{2}\left\|\mathbf{LZS}-\mathbf{Z}\right\|_{\mathrm{F}}^2+\alpha\left\|\mathbf{S}\right\|_1 \quad s.t. \quad \mathbf{S}=\mathbf{V}, \quad \pi_\Omega(\mathbf{V})=0$$

where Ω is the set containing indices of diagonal elements in similarity matrix $\mathbf{S}$ and $\pi_\Omega:\mathbb{R}^{n\times n\to n\times n}$ is a linear operator that keeps the entries in Ω unchanged and sets outside Ω, i.e., in $\bar{\Omega}$, zeros. Following the ADMM procedure, the above constrained optimization problem can be solved as a serious of unconstrained minimization problems using augmented Lagrangian function, which is presented as:

$$\mathcal{L}\left(\mathbf{S},\mathbf{V},\mathbf{B}\right)=\tfrac{\lambda}{2}\left\|\mathbf{XV}-\mathbf{X}\right\|_{\mathrm{F}}^2+\tfrac{\beta}{2}\left\|\mathbf{LZV}-\mathbf{Z}\right\|_{\mathrm{F}}^2+\alpha\left\|\mathbf{S}\right\|_1+\left\langle\mathbf{B},\mathbf{S}-\mathbf{V}\right\rangle+\tfrac{\rho}{2}\left\|\mathbf{S}-\mathbf{V}\right\|_{\mathrm{F}}^2$$

Here, ρ is the penalty parameter and $\mathbf{B}$ is the Lagrange multiplier. A sequential minimization of variables $\mathbf{S}$, $\mathbf{V}$ and $\mathbf{B}$ can be conducted by the scaled ADMM iterations:

$$\mathbf{S}_{k+1}=\mathcal{S}_{\alpha/\rho}(\mathbf{V}_k+\rho_k^{-1}\mathbf{B}_k)$$
$$\mathbf{V}_{k+1}=\pi_{\bar{\Omega}}((\mathbf{T}_1+\rho\mathbf{I})^{-1}(\mathbf{T}_2+\mathbf{B}+\rho\mathbf{S}))$$

where $\mathbf{T}_1=\lambda\mathbf{X}^\top\mathbf{X}+\beta\mathbf{Z}^\top\mathbf{L}^\top\mathbf{LZ}$ and $\mathbf{T}_2=\lambda\mathbf{X}^\top\mathbf{X}+\beta\mathbf{Z}^\top\mathbf{L}^\top\mathbf{Z}$. And $\mathcal{S}$ is the proximity operator of the ℓ_1 norm, which is defined as $S_\omega(a)=(a-\omega)_+-(-a-\omega)_+$. Then, the Lagrange multiplier matrix $\mathbf{B}$ and penalty parameter ρ are updated based on following rules:

$$\mathbf{B}_{k+1}=\mathbf{B}_k+\rho(\mathbf{S}_{k+1}-\mathbf{V}_{k+1})$$
$$\rho_{k+1}=\min(\rho_{\max},c\rho_k)$$

where $\rho_{\max}$ is the maximum value of ρ and c is a positive updating constant which is defined by users.

2) Updating $\mathbf{Z}$*:* With $\mathbf{S}$, $\mathbf{W}$ and $\mathbf{b}$ fixed, the optimization problem (3) reduces to

$$\min_{\mathbf{Z}}\left\|\mathbf{J}\circ(\mathbf{Z}-\mathbf{Y})\right\|_{\mathrm{F}}^2+\frac{\beta}{2}\left\|\mathbf{LZS}-\mathbf{Z}\right\|_{\mathrm{F}}^2+\frac{\gamma}{2}\left\|\mathbf{Z}-\mathbf{F}\right\|_{\mathrm{F}}^2 \quad (4)$$

where $\mathbf{F}=\mathbf{WX}+\mathbf{b}\mathbf{1}_n^\top$. The above optimization problem can be solve by updating $\mathbf{Z}$ with gradient decent. Specifically, the gradient of the objective function with respective to $\mathbf{Z}$ is

$$\nabla\mathbf{Z}=\mathbf{J}\circ(\mathbf{Z}-\mathbf{Y})+\gamma(\mathbf{Z}-\mathbf{F})+\beta\left(\mathbf{Z}-\mathbf{LZS}-\mathbf{L}^\top\mathbf{ZS}^\top+\mathbf{L}^\top\mathbf{LZSS}^\top\right)$$

3) Updating $\mathbf{W}$ *and* $\mathbf{b}$*:* With $\mathbf{S}$ and $\mathbf{Z}$ fixed, the optimization problem reduces to

$$\min_{\mathbf{W},\mathbf{b}}\tfrac{\gamma}{2}\mathrm{tr}\left(\mathbf{\Sigma\Sigma}^\top\right)+\tfrac{\mu}{2}\mathrm{tr}\left(\mathbf{WW}^\top\right) \quad s.t. \quad \mathbf{Z}=\mathbf{WX}+\mathbf{b}\mathbf{1}_n+\mathbf{\Sigma} \quad (5)$$

Here $\mathbf{\Sigma}=[\mathbf{e}_1,...,\mathbf{e}_n]\in\mathbb{R}^{q\times n}$, where $\mathbf{e}_i=\boldsymbol{z}_i-(\mathbf{W}\boldsymbol{x}_i+\mathbf{b})$. To kernelize our method, we introduce $\mathbf{X}'=[\boldsymbol{x}_1^{'},...,\boldsymbol{x}_n^{'}]\in\mathbb{R}^{d\times n}$, where $\boldsymbol{x}_i^{'}=\phi(\boldsymbol{x}_i)$ and $\phi(\cdot):\mathbb{R}^n\to\mathbb{R}^h$ is a feature mapping that maps the feature space to some higher (maybe infinite) dimensional Hilbert space with h dimensions. $\mathrm{tr}\left(\cdot\right)$ denotes the trace norm operator with the property $\mathrm{tr}\left(\mathbf{WW}^\top\right)=\left\|\mathbf{W}\right\|_{\mathrm{F}}^2$. To solve the above optimization problem, its Lagrangian can be formulated as:

$$\mathcal{L}\left(\mathbf{W},\mathbf{b},\mathbf{\Sigma},\mathbf{A}\right)=\frac{\gamma}{2}\mathrm{tr}\left(\mathbf{\Sigma\Sigma}^\top\right)+\frac{\mu}{2}\mathrm{tr}\left(\mathbf{WW}^\top\right)-\mathrm{tr}\left(\mathbf{A}^\top\left(\mathbf{WX}'+\mathbf{b}\mathbf{1}_n^\top+\mathbf{\Sigma}-\mathbf{Z}\right)\right)$$

where $\mathbf{A}=[\boldsymbol{\alpha}_1,...,\boldsymbol{\alpha}_n]^\top\in\mathbb{R}^{q\times n}$ is the matrix that stores the Lagrange multipliers.We now optimize out $\mathbf{W}$,$\mathbf{b}$, $\mathbf{\Sigma}$ and $\mathbf{A}$ according to the KKT conditions:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{W}}=0\Rightarrow\mathbf{W}=\frac{1}{\mu}\mathbf{AX}^{'\top},$$
$$\frac{\partial\mathcal{L}}{\partial\mathbf{b}}=0\Rightarrow\mathbf{A}\mathbf{1}_n=\mathbf{0}_q,$$
$$\frac{\partial\mathcal{L}}{\partial\mathbf{\Sigma}}=0\Rightarrow\mathbf{A}=\gamma\mathbf{\Sigma},$$
$$\frac{\partial\mathcal{L}}{\partial\mathbf{A}}=0\Rightarrow\mathbf{Z}=\mathbf{WX}^{'}+\mathbf{b}\mathbf{1}_n+\mathbf{\Sigma}.$$

Above equations can be arranged as follows:

$$\mathbf{WX}^{'}+\mathbf{b}\mathbf{1}_n^\top+\mathbf{\Sigma}=\mathbf{Z}$$
$$\frac{1}{\mu}\mathbf{AX}^{'\top}\mathbf{X}^{'}+\mathbf{b}\mathbf{1}_n^\top+\frac{1}{\gamma}\mathbf{A}=\mathbf{Z}$$

For simplicity, we introduce the positive definite matrix $\mathbf{H}=\frac{1}{\mu}\mathbf{K}+\frac{1}{\gamma}\mathbf{I}_{n\times n}$ and define $\mathbf{K}=\mathbf{X}^{'\top}\mathbf{X}^{'}\in\mathbb{R}^{n\times n}$ by its elements $k_{ij}=\phi(\boldsymbol{x}_i)^\top\phi(\boldsymbol{x}_j)=\mathcal{K}(\boldsymbol{x}_i,\boldsymbol{x}_j)$, where $\mathcal{K}(\cdot,\cdot)$ is the kernel function. Then, after several simple mathematical computations, we can obtain the final solutions:

$$\mathbf{b}=\frac{\mathbf{ZH}^{-1}\mathbf{1}_n}{\mathbf{1}_n^\top\mathbf{H}^{-1}\mathbf{1}_n}$$
$$\mathbf{A}=\left(\mathbf{Z}-\mathbf{b}\mathbf{1}_n^\top\right)\mathbf{H}^{-1}$$

IV. Experiment

A. Experimental Setting

We perform experiments on eight data sets. These data sets are related to various real-world tasks: *YeastBP*, *YeastCC* and *YeastMF* for gene function prediction, *music_style* for music recognition, *image*,*corel5K* and *corel16K* for image annotation, *delicious* for text categorization. Several characteristics about these data sets such as the *number of instances*, *number of features*, *number of class labels*,*cardinality* and *domain* are illustrated in Table I. We also conduct some pre-processing to facilitate the partially labeling as in [5], [8]. Specifically,

TABLE I
CHARACTERISTICS OF THE EXPERIMENTAL DATA SETS.

Data set	# Instances	# Features	# Class Labels	Cardinality	Domain
YeastBP	6139	6139	217	5.537	biology
YeastCC	6139	6139	50	1.348	biology
YeastMF	6139	6139	39	1.005	biology
music_style	6839	98	10	1.44	music
delicious	16105	500	983	19.020	text
image	2000	294	5	1.236	image
corel5K	5000	499	374	3.522	image
corel16k	13811	500	161	2.867	image

TABLE II
EXPERIMENTAL RESULTS OF EACH COMPARING APPROACH IN TERMS OF *ranking loss* AND *average precision* ON *YeastBP*, *YeastCC* AND *YeastMF*, WHERE •/○ INDICATES WHETHER SPPML IS SUPERIOR/INFERIOR TO THE OTHER METHODS ON EACH DATA SET (PAIR t-TEST AT 0.05 SIGNIFICANCE LEVEL).

	$p = 0.10$	$p = 0.15$	$p = 0.20$	$p = 0.30$	$p = 0.40$	$p = 0.50$
Ranking loss (the smaller, the better)						
SSPML	0.308 ± 0.011	0.283 ± 0.007	0.262 ± 0.009	0.232 ± 0.009	0.217 ± 0.010	0.203 ± 0.006
PARVLS	0.586 ± 0.042•	0.520 ± 0.022•	0.516 ± 0.030•	0.456 ± 0.032•	0.452 ± 0.025•	0.436 ± 0.023•
PARMAP	0.352 ± 0.022•	0.333 ± 0.031•	0.330 ± 0.025•	0.312 ± 0.015•	0.298 ± 0.010•	0.290 ± 0.007•
PMLLRS	0.371 ± 0.009•	0.343 ± 0.008•	0.313 ± 0.005•	0.274 ± 0.009•	0.246 ± 0.012•	0.237 ± 0.010•
PMLNI	0.355 ± 0.010•	0.339 ± 0.009•	0.305 ± 0.009•	0.297 ± 0.016•	0.386 ± 0.006•	0.380 ± 0.013•
fPML	0.497 ± 0.006•	0.476 ± 0.011•	0.464 ± 0.012•	0.456 ± 0.009•	0.446 ± 0.010•	0.446 ± 0.005•
Average precision (the greater, the better)						
SSPML	0.249 ± 0.015	0.277 ± 0.017	0.303 ± 0.016	0.342 ± 0.011	0.363 ± 0.014	0.376 ± 0.009
PARVLS	0.069 ± 0.009•	0.084 ± 0.014•	0.075 ± 0.010•	0.097 ± 0.013•	0.087 ± 0.011•	0.095 ± 0.016•
PARMAP	0.124 ± 0.014•	0.157 ± 0.039•	0.166 ± 0.029•	0.182 ± 0.020•	0.201 ± 0.032•	0.222 ± 0.031•
PMLLRS	0.196 ± 0.009•	0.228 ± 0.013•	0.254 ± 0.012•	0.291 ± 0.013•	0.327 ± 0.014•	0.344 ± 0.012•
PMLNI	0.210 ± 0.007•	0.237 ± 0.003•	0.260 ± 0.012•	0.250 ± 0.017•	0.142 ± 0.006•	0.148 ± 0.005•
fPML	0.094 ± 0.006•	0.095 ± 0.007•	0.105 ± 0.009•	0.093 ± 0.002•	0.090 ± 0.003•	0.087 ± 0.003•

	$p = 0.10$	$p = 0.15$	$p = 0.20$	$p = 0.30$	$p = 0.40$	$p = 0.50$
Ranking loss (the smaller, the better)						
SSPML	0.326 ± 0.017	0.299 ± 0.007	0.287 ± 0.022	0.252 ± 0.007	0.247 ± 0.018	0.239 ± 0.013
PARVLS	0.590 ± 0.085•	0.538 ± 0.076•	0.465 ± 0.093•	0.437 ± 0.044•	0.431 ± 0.070•	0.423 ± 0.065•
PARMAP	0.360 ± 0.041	0.358 ± 0.029•	0.349 ± 0.028•	0.341 ± 0.030•	0.326 ± 0.031•	0.320 ± 0.024•
PMLLRS	0.392 ± 0.014•	0.341 ± 0.008•	0.323 ± 0.012•	0.311 ± 0.011•	0.277 ± 0.008•	0.256 ± 0.012•
PMLNI	0.338 ± 0.016•	0.318 ± 0.010•	0.298 ± 0.005	0.293 ± 0.016•	0.284 ± 0.016•	0.273 ± 0.017•
fPML	0.504 ± 0.012•	0.433 ± 0.024•	0.436 ± 0.030•	0.446 ± 0.018•	0.419 ± 0.008•	0.411 ± 0.018•
Average precision (the greater, the better)						
SSPML	0.308 ± 0.027	0.344 ± 0.017	0.364 ± 0.017	0.398 ± 0.016	0.409 ± 0.031	0.413 ± 0.021
PARVLS	0.181 ± 0.028•	0.170 ± 0.033•	0.175 ± 0.048•	0.195 ± 0.046•	0.202 ± 0.042•	0.199 ± 0.030•
PARMAP	0.239 ± 0.031•	0.260 ± 0.027•	0.277 ± 0.038•	0.269 ± 0.026•	0.278 ± 0.027•	0.282 ± 0.009•
PMLLRS	0.243 ± 0.008•	0.261 ± 0.009•	0.294 ± 0.018•	0.332 ± 0.003•	0.371 ± 0.006•	0.389 ± 0.015•
PMLNI	0.294 ± 0.013	0.308 ± 0.016•	0.340 ± 0.019•	0.356 ± 0.015•	0.331 ± 0.013•	0.335 ± 0.015•
fPML	0.194 ± 0.005•	0.196 ± 0.009•	0.195 ± 0.015•	0.199 ± 0.011•	0.200 ± 0.012•	0.201 ± 0.012•

	$p = 0.10$	$p = 0.15$	$p = 0.20$	$p = 0.30$	$p = 0.40$	$p = 0.50$
Ranking loss (the smaller, the better)						
SSPML	0.273 ± 0.018	0.241 ± 0.014	0.226 ± 0.022	0.206 ± 0.017	0.191 ± 0.007	0.189 ± 0.016
PARVLS	0.565 ± 0.055•	0.558 ± 0.047•	0.563 ± 0.091•	0.492 ± 0.055•	0.530 ± 0.081•	0.471 ± 0.058•
PARMAP	0.407 ± 0.051•	0.368 ± 0.020•	0.345 ± 0.035•	0.322 ± 0.018•	0.320 ± 0.025•	0.306 ± 0.025•
PMLLRS	0.346 ± 0.025•	0.327 ± 0.009•	0.283 ± 0.009•	0.241 ± 0.011•	0.220 ± 0.016•	0.203 ± 0.008•
PMLNI	0.320 ± 0.015•	0.302 ± 0.028•	0.264 ± 0.006•	0.230 ± 0.007•	0.225 ± 0.021•	0.219 ± 0.016•
fPML	0.486 ± 0.015•	0.449 ± 0.018•	0.425 ± 0.013•	0.398 ± 0.009•	0.389 ± 0.019•	0.402 ± 0.013•
Average precision (the greater, the better)						
SSPML	0.402 ± 0.020	0.445 ± 0.023	0.468 ± 0.030	0.500 ± 0.025	0.527 ± 0.012	0.537 ± 0.026
PARVLS	0.146 ± 0.018•	0.159 ± 0.031•	0.161 ± 0.039•	0.172 ± 0.029•	0.159 ± 0.027•	0.187 ± 0.028•
PARMAP	0.231 ± 0.036•	0.246 ± 0.015•	0.272 ± 0.045•	0.307 ± 0.024•	0.309 ± 0.035•	0.330 ± 0.040•
PMLLRS	0.329 ± 0.019•	0.362 ± 0.009•	0.399 ± 0.016•	0.455 ± 0.021•	0.504 ± 0.020•	0.538 ± 0.018
PMLNI	0.372 ± 0.022•	0.402 ± 0.033•	0.437 ± 0.011•	0.476 ± 0.017•	0.493 ± 0.032•	0.491 ± 0.027•
fPML	0.209 ± 0.008•	0.224 ± 0.009•	0.226 ± 0.009•	0.249 ± 0.013•	0.249 ± 0.031•	0.206 ± 0.011•

for data sets with too many labels (more than 100 in our experiment), their rare labels are filtered out to keep under 15 labels, and instances without any relevant labels are filtered out.

There are different criteria for evaluating the performances of multi-label learning. In our experiment, we employ five

TABLE III

EXPERIMENTAL RESULTS OF EACH COMPARING APPROACH IN TERMS OF *average precision* ON *music_style*, WHERE •/○ INDICATES WHETHER SPPML IS SUPERIOR/INFERIOR TO THE OTHER METHODS ON EACH DATA SET (PAIR t-TEST AT 0.05 SIGNIFICANCE LEVEL).

	$p = 0.10$	$p = 0.15$	$p = 0.20$	$p = 0.30$	$p = 0.40$	$p = 0.50$
Ranking loss (the smaller, the better)						
SSPML	0.204 ± 0.004	0.183 ± 0.007	0.168 ± 0.012	0.159 ± 0.010	0.152 ± 0.005	0.145 ± 0.003
PARVLS	0.206 ± 0.010	0.203 ± 0.009•	0.190 ± 0.005•	0.190 ± 0.005•	0.180 ± 0.006•	0.180 ± 0.008•
PARMAP	0.180 ± 0.008○	0.183 ± 0.010	0.167 ± 0.006	0.161 ± 0.003	0.156 ± 0.005•	0.156 ± 0.005•
PMLLRS	0.214 ± 0.013	0.203 ± 0.009•	0.200 ± 0.006•	0.186 ± 0.004•	0.174 ± 0.004•	0.168 ± 0.008•
PMLNI	0.239 ± 0.010•	0.207 ± 0.005•	0.181 ± 0.007•	0.167 ± 0.008•	0.157 ± 0.006•	0.149 ± 0.008
fPML	0.192 ± 0.011○	0.176 ± 0.008○	0.164 ± 0.007	0.156 ± 0.005	0.155 ± 0.004•	0.153 ± 0.004•
Average precision (the greater, the better)						
SSPML	0.646 ± 0.005	0.682 ± 0.014	0.701 ± 0.017	0.715 ± 0.010	0.723 ± 0.006	0.729 ± 0.003
PARVLS	0.690 ± 0.004○	0.693 ± 0.008○	0.704 ± 0.007	0.702 ± 0.006•	0.705 ± 0.003•	0.704 ± 0.004•
PARMAP	0.676 ± 0.013○	0.681 ± 0.009	0.697 ± 0.007	0.704 ± 0.004•	0.708 ± 0.007•	0.710 ± 0.008•
PMLLRS	0.661 ± 0.009○	0.667 ± 0.015•	0.666 ± 0.007•	0.673 ± 0.005•	0.683 ± 0.006•	0.689 ± 0.010•
PMLNI	0.599 ± 0.009•	0.643 ± 0.008•	0.678 ± 0.014•	0.703 ± 0.008•	0.718 ± 0.009•	0.725 ± 0.009
fPML	0.679 ± 0.011○	0.688 ± 0.009	0.695 ± 0.011	0.706 ± 0.014	0.705 ± 0.006•	0.709 ± 0.006•

TABLE IV

EXPERIMENTAL RESULTS OF EACH COMPARING APPROACH IN TERMS OF *average precision* ON *delicious*, WHERE •/○ INDICATES WHETHER SPPML IS SUPERIOR/INFERIOR TO THE OTHER METHODS ON EACH DATA SET (PAIR t-TEST AT 0.05 SIGNIFICANCE LEVEL).

	$p = 0.10$	$p = 0.15$	$p = 0.20$	$p = 0.30$	$p = 0.40$	$p = 0.50$
Ranking loss (the smaller, the better)						
SSPML	0.289 ± 0.004	0.275 ± 0.004	0.268 ± 0.005	0.260 ± 0.006	0.256 ± 0.002	0.254 ± 0.002
PARVLS	0.352 ± 0.006•	0.342 ± 0.007•	0.328 ± 0.006•	0.315 ± 0.004•	0.305 ± 0.003•	0.303 ± 0.004•
PARMAP	0.306 ± 0.005•	0.305 ± 0.005•	0.298 ± 0.002•	0.291 ± 0.002•	0.287 ± 0.004•	0.284 ± 0.004•
PMLLRS	0.291 ± 0.005	0.283 ± 0.004•	0.279 ± 0.004•	0.275 ± 0.004•	0.273 ± 0.003•	0.271 ± 0.002•
PMLNI	0.285 ± 0.004	0.281 ± 0.004•	0.276 ± 0.003•	0.274 ± 0.003•	0.272 ± 0.003•	0.273 ± 0.002•
fPML	0.296 ± 0.003•	0.291 ± 0.004•	0.289 ± 0.004•	0.285 ± 0.004•	0.283 ± 0.002•	0.281 ± 0.003•
Average precision (the greater, the better)						
SSPML	0.587 ± 0.004	0.602 ± 0.003	0.609 ± 0.003	0.618 ± 0.006	0.621 ± 0.006	0.622 ± 0.002
PARVLS	0.564 ± 0.008•	0.573 ± 0.008•	0.582 ± 0.005•	0.595 ± 0.007•	0.603 ± 0.005•	0.606 ± 0.007•
PARMAP	0.559 ± 0.008•	0.557 ± 0.006•	0.564 ± 0.004•	0.570 ± 0.007•	0.573 ± 0.006•	0.577 ± 0.007•
PMLLRS	0.582 ± 0.005•	0.593 ± 0.004•	0.597 ± 0.005•	0.601 ± 0.005•	0.603 ± 0.005•	0.605 ± 0.002•
PMLNI	0.590 ± 0.006	0.596 ± 0.003•	0.601 ± 0.004•	0.603 ± 0.005•	0.605 ± 0.004•	0.604 ± 0.002•
fPML	0.579 ± 0.005•	0.585 ± 0.005•	0.587 ± 0.004•	0.589 ± 0.007•	0.591 ± 0.003•	0.594 ± 0.004•

commonly used criteria including *hamming loss*, *ranking loss*, *one error*, *coverage* and *average precision*. For *hamming loss*, *ranking loss*, *one error* and *coverage* metrics, the smaller value, the better the performance; for *average precision* metric, the greater value, the better the performance. More detail about theses evaluation metrics can be found in [1].

To demonstrate the effectiveness of the proposed SSPML method, we compare it with five state-of-the art PML algorithms as follows:

- fPML [7]. It employ the low-rank approximation of the observed instance-label association matrix to estimate the labeling confidence and then trains multi-label classifier.
- PARTICLE [8]. It transforms the PML task into a multi-label problem through a label propagation procedure. Then a calibrated label ranking model is induced to instantiate two PML methods PAR-VLS and PAR-MAP.
- PML-LRS [10]. It utilizes low-rank and sparse decomposition scheme to capture the ground-truth label matrix and irrelevant label matrix from the observed candidate label matrix.
- PML-NI [6]. It jointly learns a noisy label identifier, which identifies feature-induced noisy labels, as well as a multi-label classifier for prediction.

For the comparing methods, parameters are set as suggested in the original paper. Specifically, for fPML, balancing parameters are set as $\lambda_1 = 0.1$, $\lambda_2 = 1$ and $\lambda_3 = 10$. For PAR-VLS and PAR-MAP, balancing parameter $\alpha = 0.95$ and credible label elicitation threshold $thr = 0.9$. For PML-LRS, balancing parameters are set as $\gamma = 0.01$, $\beta = 0.1$ and $\eta = 1$. For PML-NI, balancing parameters are set as $\lambda = 1$, $\beta = 1$ and $\gamma = 0.5$. For our SSPML method, balancing parameters are set as $\lambda = 1$, $\beta = 1$, $\gamma = 1$, $\mu = 1$ and $\alpha = 0.1$.

To construct partial multi-label assignment for training data of each dataset except for the first four real-world PML data sets, for each example $\boldsymbol{x}_i$, we randomly add the irrelevant noisy labels of $\boldsymbol{x}_i$ with $\theta\%$ number of ground-truth labels, and $\theta\%$ is also randomly assigned by one of $\{50\%, 100\%, 150\%\}$. For each data set, we consider the percentage of partially labeled examples in the whole training set by randomly sampling $p \in \{0.1, 0.15, 0.2, 0.3, 0.4, 0.5\}$ instances from the whole training set with their candidate sets and the others without any supervised information. For comparing methods, only sampled partial label examples and their candidate label sets are provided due to the fact that PML methods cannot utilize the unlabeled data.

TABLE V

EXPERIMENTAL RESULTS OF EACH COMPARING APPROACH IN TERMS OF *average precision* ON *image*, *corek5K* AND *corel16K*, WHERE •/○ INDICATES WHETHER SPPML IS SUPERIOR/INFERIOR TO THE OTHER METHODS ON EACH DATA SET (PAIR t-TEST AT 0.05 SIGNIFICANCE LEVEL).

	$p = 0.10$	$p = 0.15$	$p = 0.20$	$p = 0.30$	$p = 0.40$	$p = 0.50$
Ranking loss (the smaller, the better)						
SSPML	0.274 ± 0.015	0.238 ± 0.016	0.224 ± 0.009	0.196 ± 0.014	0.190 ± 0.016	0.188 ± 0.018
PARVLS	0.348 ± 0.048•	0.343 ± 0.036•	0.334 ± 0.035•	0.299 ± 0.039•	0.285 ± 0.025•	0.280 ± 0.042•
PARMAP	0.357 ± 0.035•	0.344 ± 0.027•	0.378 ± 0.031•	0.348 ± 0.036•	0.351 ± 0.039•	0.350 ± 0.025•
PMLLRS	0.488 ± 0.023•	0.516 ± 0.007•	0.531 ± 0.026•	0.457 ± 0.019•	0.427 ± 0.015•	0.359 ± 0.019•
PMLNI	0.498 ± 0.011•	0.525 ± 0.008•	0.517 ± 0.019•	0.462 ± 0.016•	0.414 ± 0.029•	0.375 ± 0.014•
fPML	0.481 ± 0.017•	0.512 ± 0.010•	0.517 ± 0.024•	0.463 ± 0.020•	0.425 ± 0.017•	0.378 ± 0.008•
Average precision (the greater, the better)						
SSPML	0.691 ± 0.017	0.725 ± 0.011	0.740 ± 0.006	0.767 ± 0.019	0.777 ± 0.018	0.778 ± 0.016
PARVLS	0.648 ± 0.033•	0.649 ± 0.046•	0.663 ± 0.030•	0.698 ± 0.023•	0.710 ± 0.030•	0.710 ± 0.034•
PARMAP	0.635 ± 0.023•	0.652 ± 0.025•	0.646 ± 0.036•	0.652 ± 0.045•	0.654 ± 0.028•	0.652 ± 0.021•
PMLLRS	0.491 ± 0.021•	0.477 ± 0.006•	0.474 ± 0.016•	0.539 ± 0.014•	0.554 ± 0.012•	0.619 ± 0.020•
PMLNI	0.495 ± 0.010•	0.477 ± 0.006•	0.491 ± 0.018•	0.540 ± 0.013•	0.575 ± 0.029•	0.609 ± 0.013•
fPML	0.491 ± 0.010•	0.477 ± 0.013•	0.478 ± 0.010•	0.535 ± 0.016•	0.556 ± 0.018•	0.609 ± 0.011•

	$p = 0.10$	$p = 0.15$	$p = 0.20$	$p = 0.30$	$p = 0.40$	$p = 0.50$
Ranking loss (the smaller, the better)						
SSPML	0.332 ± 0.007	0.296 ± 0.020	0.275 ± 0.008	0.250 ± 0.003	0.240 ± 0.006	0.230 ± 0.004
PARVLS	0.623 ± 0.097•	0.559 ± 0.047•	0.492 ± 0.036•	0.420 ± 0.021•	0.404 ± 0.020•	0.385 ± 0.017•
PARMAP	0.317 ± 0.008○	0.314 ± 0.006•	0.310 ± 0.009•	0.307 ± 0.004•	0.303 ± 0.005•	0.294 ± 0.009•
PMLLRS	0.307 ± 0.012○	0.289 ± 0.009	0.274 ± 0.010	0.253 ± 0.014	0.242 ± 0.009	0.234 ± 0.009
PMLNI	0.276 ± 0.015○	0.265 ± 0.010○	0.252 ± 0.003○	0.241 ± 0.011	0.284 ± 0.005•	0.229 ± 0.006
fPML	0.316 ± 0.007○	0.313 ± 0.008•	0.310 ± 0.004•	0.298 ± 0.003•	0.304 ± 0.010•	0.283 ± 0.010•
Average precision (the greater, the better)						
SSPML	0.426 ± 0.006	0.459 ± 0.019	0.485 ± 0.007	0.500 ± 0.009	0.511 ± 0.008	0.518 ± 0.011
PARVLS	0.370 ± 0.016•	0.383 ± 0.019•	0.376 ± 0.024•	0.407 ± 0.017•	0.404 ± 0.020•	0.403 ± 0.020•
PARMAP	0.419 ± 0.004•	0.418 ± 0.007•	0.421 ± 0.006•	0.427 ± 0.006•	0.432 ± 0.008•	0.438 ± 0.009•
PMLLRS	0.421 ± 0.016	0.440 ± 0.006•	0.461 ± 0.016•	0.479 ± 0.016•	0.493 ± 0.012•	0.503 ± 0.013•
PMLNI	0.465 ± 0.012○	0.479 ± 0.017○	0.489 ± 0.010	0.503 ± 0.014	0.426 ± 0.011•	0.508 ± 0.008•
fPML	0.417 ± 0.007•	0.420 ± 0.010•	0.424 ± 0.009•	0.437 ± 0.006•	0.424 ± 0.014•	0.455 ± 0.004•

	$p = 0.10$	$p = 0.15$	$p = 0.20$	$p = 0.30$	$p = 0.40$	$p = 0.50$
Ranking loss (the smaller, the better)						
SSPML	0.305 ± 0.005	0.270 ± 0.006	0.254 ± 0.003	0.240 ± 0.005	0.235 ± 0.005	0.231 ± 0.004
PARVLS	0.487 ± 0.037•	0.572 ± 0.038•	0.426 ± 0.015•	0.406 ± 0.009•	0.389 ± 0.012•	0.395 ± 0.009•
PARMAP	0.317 ± 0.004•	0.303 ± 0.003•	0.295 ± 0.007•	0.294 ± 0.004•	0.277 ± 0.006•	0.270 ± 0.007•
PMLLRS	0.269 ± 0.007○	0.256 ± 0.004○	0.250 ± 0.004	0.241 ± 0.007	0.237 ± 0.003	0.234 ± 0.006•
PMLNI	0.263 ± 0.004○	0.280 ± 0.015	0.248 ± 0.003	0.265 ± 0.003•	0.236 ± 0.004	0.232 ± 0.008
fPML	0.313 ± 0.009•	0.314 ± 0.008•	0.293 ± 0.006•	0.299 ± 0.002•	0.267 ± 0.004•	0.260 ± 0.008•
Average precision (the greater, the better)						
SSPML	0.431 ± 0.006	0.454 ± 0.006	0.467 ± 0.004	0.477 ± 0.005	0.479 ± 0.008	0.482 ± 0.009
PARVLS	0.381 ± 0.014•	0.402 ± 0.006•	0.405 ± 0.015•	0.420 ± 0.006•	0.426 ± 0.007•	0.422 ± 0.007•
PARMAP	0.405 ± 0.003•	0.417 ± 0.009•	0.419 ± 0.008•	0.427 ± 0.008•	0.441 ± 0.005•	0.447 ± 0.006•
PMLLRS	0.440 ± 0.005○	0.453 ± 0.005	0.457 ± 0.005•	0.467 ± 0.006•	0.471 ± 0.008•	0.473 ± 0.008•
PMLNI	0.450 ± 0.005○	0.435 ± 0.014•	0.462 ± 0.003•	0.445 ± 0.004•	0.471 ± 0.010•	0.476 ± 0.006•
fPML	0.417 ± 0.009•	0.418 ± 0.008•	0.433 ± 0.009•	0.433 ± 0.003•	0.456 ± 0.003•	0.460 ± 0.010•

TABLE VI

FRIEDMAN STATISTICS F_F IN TERMS OF EACH EVALUATION METRIC AND THE CRITICAL VALUE AT 0.05 SIGNIFICANCE LEVEL (# COMPARING ALGORITHMS $k = 6$, # DATA SETS $N = 48$).

Evaluation metric	F_F	critical value
Hamming Loss	63.7239	
Ranking loss	54.9085	
One Error	26.7657	2.2946
Coverage	34.6036	
Average Precision	47.8730	

B. Comparison Results

Due to the page limit, we report the statistical summary results for all of the five performance measures, while only report detailed results of each comparing methods in terms of *ranking loss* and *average precision*. The results on *ranking loss* and *average precision* are reported in Table II, III, V and IV, while similar results can be observed in terms of other evaluation metrics. For each data set, pairwise t-test based on five-fold cross validation (at 0.05 significance level) is conducted to show whether the performance of SSPML is significantly different to the comparing approaches. For gene function prediction tasks, the results reported in Table II show that SSPML outperforms comparing methods with significant superiority. Accordingly, it can be observed that: 1) SSPML achieves better performances than comparing methods in all cases on *Yeast BP*; 2) SSPML outperforms comparing methods significantly in almost all cases except on *Yeast CC*

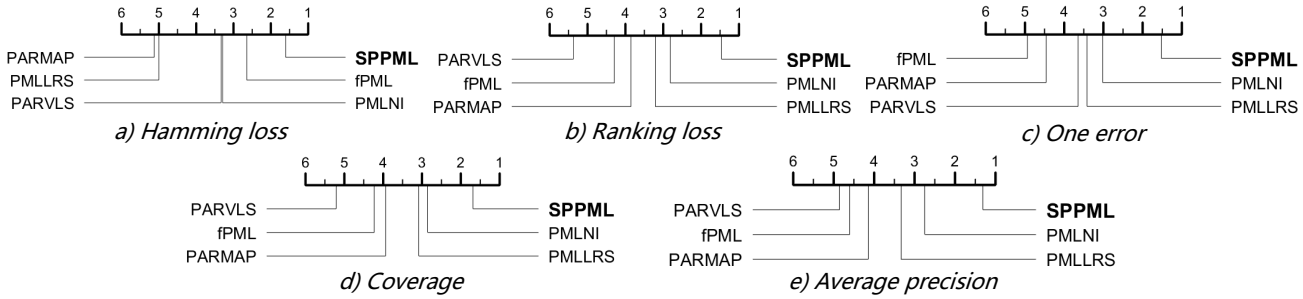


Fig. 2. Comparison of PML-NI (control algorithm) against five comparing algorithms with the Bonferroni-Dunn test. Algorithms not connected with PML-NI in the CD diagram are considered to have a significantly different performance from the control algorithm (CD = 0.9837 at 0.05 significance level).

for case with sampling rate $p = 0.10$, where PARMAP and PMLNI achieve comparable performances with SSPML in terms of *ranking loss* and *average precision*, respectively; 3) SSPML outperforms comparing methods significantly in almost all cases on *YeastMF* except for case with sampling rate $p = 0.50$, where PMLLRS is comparable to SSPML in terms of *average precision*. For music recognition tasks, the results shown in Table III demonstrate that SSPML achieves highly comparable performances to all the state-of-art PML methods. Based on the experimental results, it can be observed that SSPML is comparable or significantly better than the comparing methods in most cases on *music_style* except for cases with sampling rate $p = 0.10$ and $p = 0.15$, where SSPML is comparable or worse than the comparing methods. One possible reason is that there is no enough examples for providing SSPML with adequate supervised information when p is small. For text categorization tasks, as shown in Table IV, SSPML significantly outperforms other methods in almost all cases on *delicious* except for case with sampling rate $p = 0.10$, where PML-LRS and PML-NI are comparable to SPPML. For image annotation tasks, results are reported on Table V, from which we can observe that SSPML also achieves highly comparable performances to all the compared methods. Based on the experimental results, it can be observed that: 1) SSPML outperforms comparing methods significantly in all cases in terms of *ranking loss* and *average precision* on *image*. 2) SSPML achieves better performances than PARVLS, PARMAP and fPML in almost all cases on *corel5k* and *corel16k* except for the case with sampling rate $p = 0.10$, where PARMAP and fPML outperforms SSPML in terms of *ranking loss* on *corel5K*; 3) SSPML achieves better or comparable performance than PMLLRS and PMLNI except for cases with sampling rate $p = 0.10$ and $p = 0.15$, where PMLLRS and PMLNI show some superiority in some cases.

Furthermore, *Friedman test* [32] is employed as the statistic test to evaluate the relative performance among the comparing methods. Assume that there are k algorithms and N data sets. Let r_i^j denotes the rank of j-th algorithm on the i-th data set. The average ranks of algorithms $R_j = \frac{1}{N}\sum_i r_i^j$ is used for Friedman test comparison. Under the null-hypothesis, which indicates that all the algorithms have equivalent performance, the Friedman statistic $F_F$with respective to the F-distribution with $(k-1)(N-1)$ degree of freedom can be defined:

$$F_F = \frac{(N-1)\chi_F^2}{N(k-1)-\chi_F^2} \tag{6}$$

where,

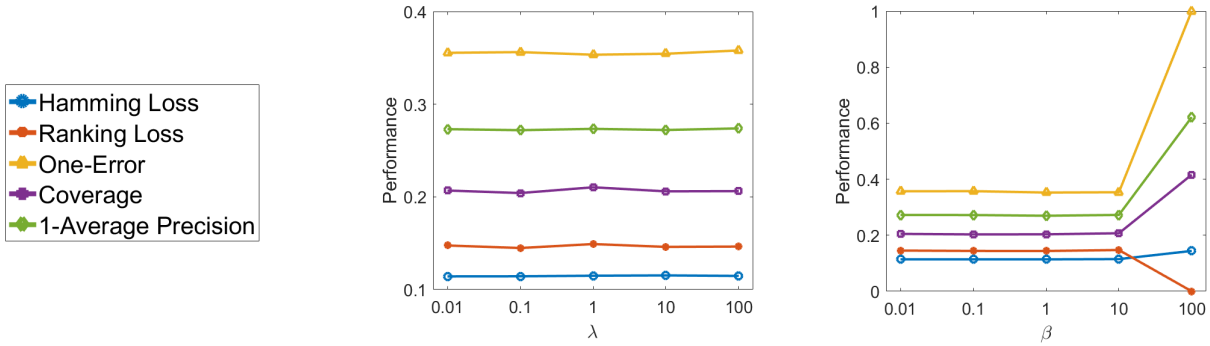
$$\chi_F^2 = \frac{12N}{k(k-1)}\left[\sum_j R_j^2 - \frac{k(k+1)^2}{4}\right] \tag{7}$$

Table VI presents the Friedman statistics F_F and the corresponding critical value with respective to each evaluation metric. For each evaluation metric, the null hypothesis of indistinguishable performance among the comparing algorithm is rejected at 0.05 significance level.

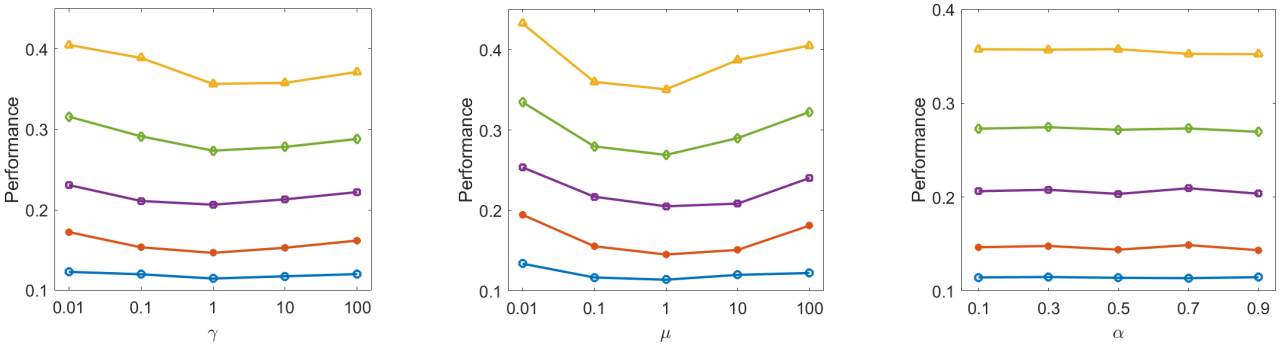
In final, we use the post-hoc *Bonferroni-Dunn test* [32] to evaluate the relative performance among comparing methods. Here, PML-NI is regarded as the control method whose average rank difference against the comparing algorithm is calibrated with the *critical difference* (CD):

$$CD = q_\alpha\sqrt{\frac{k(k+1)}{6N}} \tag{8}$$

where critical value $q_\alpha = 2.576$ at 0.05 significance level. Accordingly, SPPML is deemed to have significantly different performance to one comparing algorithm if their average ranks differ by at least one CD (CD = 0.9837 in our experiment: # comparing algorithms $k = 6$, # data sets $N = 8 \times 6 = 48$). Figure 2 shows the CD diagrams [32] on each evaluation metric, where the average rank of each comparing algorithm is marked along the axis (lower ranks to the right). In each subfigure, any comparing algorithms whose average rank is within one CD to that of SSPML is interconnected to each other with a thick line. It can be observed that SSPML achieves the best (lowest) average rank among comparing methods and outperforms all other comparing methods at least one CD in terms of all evaluation metrics. The experimental results demonstrate the significance of the superiority for our SSPML approach.



(a) Legend (b) performance curve with λ changes (c) performance curve with β changes



(d) performance curve with γ changes (e) performance curve with μ changes (f) performance curve with α changes

Fig. 3. Results of PML-NI with varying value of trade-off parameters on *music_style*.

C. Sensitive Analysis

In this section, we study the influences of five balancing parameters, λ, β, γ, μ and α for the proposed approach on the real-world data sets. We conducted experiments by varying one parameter while keeping the other four parameters fixed. Due to the page limit, we only show the experimental results which are measured by the five evaluation metrics on real-world data set *music_style* in Figure 3. As we can see, in general, performance is not sensitive to the parameters except for the parameter β, whose performance will be significantly degraded when the value of β is too large (approximates to 100 in the experiment). Therefore we can safely set them in a wide range in practice.

V. Conclusion

In this paper, we propose a new learning framework named semi-supervised partial multi-label learning (SSPML), where each instance is either associated with a candidate label set or even without any supervised information. A latent label variable vector is maintained for each instance as the low-dimensional embedding of the feature space. By minimizing the sparse reconstruction error, label variables along with similarity weights are optimized by sharing consistent similarity measurement between the feature space and label space. Meanwhile, label variables are employed to induce a classifier for semi-supervised multi-label prediction. Experiments are performed on multiple datasets from various applications; and results validate that the proposed approach are superior to state-of-the-art partial multi-label approaches. In the future, we plan to improve the SSPML algorithms by exploiting other structure information of unlabeled data.

VI. Acknowledgment

This research was supported by the Fundamental Research Funds for the Central Universities (NE2019104) and the China University S&T Innovation Plan Guided by the Ministry of Education.

References

[1] M. Zhang and Z. Zhou, "A review on multi-label learning algorithms," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 8, pp. 1819–1837, 2014.

[2] D. Turnbull, L. Barrington, D. A. Torres, and G. R. G. Lanckriet, "Semantic annotation and retrieval of music and sound effects," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 16, no. 2, pp. 467–476, 2008.

[3] G. Tsoumakas, I. Katakis, Vlahavas, and Ioannis, "Effective and efficient multilabel classification in domains with large number of labels," *Mining Multidimensional Data*, 2008.

[4] Z.-M. Chen, X.-S. Wei, P. Wang, and Y. Guo, "Multi-label image recognition with graph convolutional networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 5177–5186.

[5] M. Xie and S. Huang, "Partial multi-label learning," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018, pp. 4302–4309.

[6] M.-K. Xie and S.-J. Huang, "Partial multi-label learning with noisy label identification," in *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, (AAAI-20)*, 2020.

[7] Q. Zhang, Y. Zhong, and M. Zhang, "Feature-induced labeling information enrichment for multi-label learning," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18)*, 2018, pp. 4446–4453.

[8] J. Fang and M. Zhang, "Partial multi-label learning via credible label elicitation," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, (AAAI-19)*, 2019.

[9] Y. Yan and Y. Guo, "Adversarial partial multi-label learning," *arXiv preprint arXiv:1909.06717*, 2019.

[10] T. W. C. L. Lijuan Sun, Songhe Feng and Y. Jin, "Partial multi-label learning by low-rank and sparse decomposition," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence, (AAAI-19)*, 2019.

[11] H. Wang, W. Liu, Y. Zhao, C. Zhang, T. Hu, and G. Chen, "Discriminative and correlative partial multi-label learning," in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019*, 2019, pp. 3691–3697.

[12] Q.-G. C. Y. H. Ze-Sen Chen, Xuan Wu and M.-L. Zhang, "Multi-view partial multi-label learning with graph-based disambiguation," in *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence, (AAAI-20)*, 2020.

[13] S. He, K. Deng, L. Li, S. Shu, and L. Liu, "Discriminatively relabel for partial multi-label learning," in *2019 IEEE International Conference on Data Mining (ICDM)*. IEEE, 2019, pp. 280–288.

[14] G. Lyu, S. Feng, and Y. Li, "Partial multi-label learning via probabilistic graph matching mechanism," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 105–113.

[15] J.-H. Wu, X. Wu, Q.-G. Chen, Y. Hu, and M.-L. Zhang, "Feature-induced manifold disambiguation for multi-view partial multi-label learning," in *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2020, pp. 557–565.

[16] N. Xu, Y.-P. Liu, and X. Geng, "Partial multi-label learning with label distribution." in *AAAI*, 2020, pp. 6510–6517.

[17] T. Yu, G. Yu, J. Wang, and M. Guo, "Partial multi-label learning with label and feature collaboration," *arXiv*, pp. arXiv–2003, 2020.

[18] Z. Li, G. Lyu, and S. Feng, "Partial multi-label learning via multi-subspace representation," in *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 2020, pp. 2612–2618.

[19] Y. Guo and D. Schuurmans, "Semi-supervised multi-label classification - A simultaneous large-margin, subspace learning approach," in *Machine Learning and Knowledge Discovery in Databases - European Conference, ECML PKDD 2012*, 2012, pp. 355–370.

[20] L. Wu and M. Zhang, "Multi-label classification with unlabeled data: An inductive approach," in *Asian Conference on Machine Learning, ACML 2013*, 2013, pp. 197–212.

[21] Z. Erich, "A framework for active learning of beam alignment in vehicular millimeter wave communications by onboard sensors," *ZTE Communications*, vol. 17, no. 2, pp. 2–9, 2019.

[22] M. Gönen, "Coupled dimensionality reduction and classification for supervised and semi-supervised multilabel learning," *Pattern Recognition Letters*, vol. 38, pp. 132–141, 2014.

[23] W. Zhan and M. Zhang, "Inductive semi-supervised multi-label learning with co-training," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1305–1314.

[24] Y. Xing, G. Yu, C. Domeniconi, J. Wang, and Z. Zhang, "Multi-label co-training," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018*, 2018, pp. 2882–2888.

[25] L. Wang, Y. Liu, C. Qin, G. Sun, and Y. Fu, "Dual relation semi-supervised multi-label learning," in *Proc. AAAI*, 2020.

[26] H.-C. Dong, Y.-F. Li, and Z.-H. Zhou, "Learning from semi-supervised weak-label data," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[27] Z.-S. Chen, X. Wu, Q.-G. Chen, Y. Hu, and M.-L. Zhang, "Multi-view partial multi-label learning with graph-based disambiguation," in *Proceedings of the 34th AAAI Conference on Artificial Intelligence*, 2020.

[28] Y. Zhu, J. T. Kwok, and Z.-H. Zhou, "Multi-label learning with global and local label correlation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 6, pp. 1081–1094, 2017.

[29] G. Tsoumakas, I. Katakis, and I. Vlahavas, "Random k-labelsets for multilabel classification," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 7, pp. 1079–1089, 2011.

[30] G. Tsoumakas, A. Dimou, E. Spyromitros, V. Mezaris, I. Kompatsiaris, and I. Vlahavas, "Correlation-based pruning of stacked binary relevance models for multi-label learning," in *Proceedings of the 1st international workshop on learning from multi-label data*, 2009, pp. 101–116.

[31] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein *et al.*, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine learning*, vol. 3, no. 1, pp. 1–122, 2011.

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