Neurocomputing 260 (2017) 192–202

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Semi-supervised multi-label classification using incomplete label information

Qiaoyu Tan, Yanming Yu, Guoxian Yu, Jun Wang*

College of Computer and Information Science, Southwest University, Chongqing 400715, China

ARTICLE INFO

Article history:
Received 31 July 2016
Revised 26 February 2017
Accepted 13 April 2017
Available online 25 April 2017

Communicated by Jiayu Zhou

Keywords:
Multi-label learning
Semi-supervised learning
Incomplete labels
Label correlation

ABSTRACT

Classifying multi-label instances using incompletely labeled instances is one of the fundamental tasks in multi-label learning. Most existing methods regard this task as supervised weak-label learning problem and assume sufficient partially labeled instances are available. However, collecting or annotating such instances is expensive and time-consuming. In contrast, abundant unlabeled instances are easy to accumulate. Recently, some methods move toward exploiting unlabeled instances and performing transductive multi-label classification. However, these methods can not directly apply to new instances, which are not available during training process. In this paper, we proposed an approach called *S*emi-supervised *m*ulti-label classification using *i*ncomplete *l*abel information (SMILE for short). SMILE first estimates label correlation from partially labeled instances and replenishes missing labels of these instances. Then, it takes advantage of labeled and unlabeled instances to construct a neighborhood graph. Next, the known labels and replenished ones of labeled instances, along with unlabeled instances are exploited to train a graph based semi-supervised linear classifier. SMILE can further replenish the missing labels of training instances based on the adopted neighborhood graph. In addition, it can directly predict the labels of completely unlabeled new instances. The empirical study on multi-label datasets shows that SMILE performs significantly better than other related methods across various evaluation criteria and it is important to leverage unlabeled data with label correlation for multi-label classification.

1. Introduction

In multi-label learning task [1,2], each instance can be assigned with a set of labels. For example, a gene can be annotated with multiple functional labels [3], such as metabolism and protein synthesis; an article can include multiple topics [4], such as politics, economics, sports and cultures. A straightforward solution for multi-label learning is to transform the task into multiple binary relevance classifiers, one label for one classifier. However, such transformation limits the effectiveness by ignoring label correlation among labels. In practice, label correlation can facilitate the process of multi-label learning [1,2].

Various techniques have been proposed to exploit label correlations. Read et al. [5] introduced a classifier chain method. This method links multiple binary classifiers along a chain, each classifier in the chain deals with a binary relevant problem for a particular label and the feature space of each link in the chain is extended with the 0/1 label associations of all previous classifiers. Each chain of binary classifiers has different effect on the classification. Given that, this method randomly runs multiple chains with different orders of binary classifiers to produce an ensemble classifier chain. Zhang and Zhang [6] used Bayesian network structure to efficiently encode the conditional dependencies of labels as well as the feature set, with the feature set as the common parent of all labels. Tsoumakas et al. [7] firstly broke the initial set of labels into a number of small random subsets and trained a classifier for each subset, and then integrated these classifiers for prediction. Huang and Zhou [8] proposed a method called multi-label learning by exploiting label correlations locally (ML-LOC). ML-LOC first derives a local correlation code to enhance the feature representation of each instance, then incorporates the global discrimination fitting and local correlation sensitivity into a unified framework for multi-label classification. Indeed, intensive research has been doing in utilizing label correlation and multi-label classification. A comprehensive coverage of them is beyond the scope of this paper, reader can refer to [1] and [2], and references therein.

All these aforementioned multi-label methods are supervised approaches that often ask for sufficient labeled instances. However, sufficient labeled instances are generally difficult and expensive to accumulate, but unlabeled instances are easy to accumulate.

* Corresponding author.
E-mail address: kingjun@swu.edu.cn (J. Wang).

http://dx.doi.org/10.1016/j.neucom.2017.04.033
0925-2312/

To improve the performance, it is necessary to develop techniques that can leverage limited labeled instances and many unlabeled instances. Semi-supervised learning [9], is among one of these techniques that can take advantage of labeled and unlabeled instances. Given that, semi-supervised multi-label classification is widely studied in recent literature. Zha et al. [10] proposed a graph-based multi-label learning method. This method constructs a graph at first, in which each weighted edge represents the similarity of two nodes. Next, it simultaneously exploits the inherent correlations and label consistency over the graph, and trains a semi-supervised classifier. Kong et al. [11] proposed a transductive multi-label classifier called Tram. Tram introduces the label concept composition and assumes similar instances should have similar label concept composition. It formulates the transductive multi-label classification as an optimization problem of estimating label concept compositions and develops a closed-form solution. Guo and Schuurmans [12] proposed a transductive approach, which exploits the dependence structure between labels using large margin training algorithm [13], and adapts a subspace representation learning algorithm [14] to employ unlabeled instances. Wu and Zhang [15] proposed a semi-supervised classifier, which considers the ranking ability on labeled instances, and made use of labeled instances via maximum margin assumption. Next, they exploited unlabeled instances via appropriate regularization term, and optimized the classifier as an non-convex optimization problem. Jing et al. [16] utilized a low-rank mapping from feature space to label space via the alternating direction method of multipliers [17], and proposed a low-rank mapping based multi-label learning method. These semi-supervised multi-label methods have improved performance than using scarce labeled instances alone.

All these aforementioned methods assume the available labels of labeled instances are complete without missing. In practical scenarios, however, we may just know a subset of labels of an instance, and whether the instance should be annotated with other labels is unknown. In other words, instances are incompletely annotated [18–20]. This multi-label learning using incompletely annotated instances is called *multi-label learning with weak label* [21]. To handle that more challenging scenario, some transductive multi-label weak label learning algorithms have been proposed to replenish the missing labels of incompletely labeled instances or to predict the labels of completely unlabeled training instances [22–25]. Different from these transductive solutions, we target at predicting the labels of completely unlabeled new instances, which are not available in the training period. For this purpose, we propose an approach called semi-supervised multi-label classification using Incomplete Label information (SMILE for short). SMILE firstly replenishes the missing labels of training instances based on the label correlation estimated from incompletely labeled instances. Next, a neighborhood graph is constructed to leverage labeled and unlabeled instances. After that, SMILE takes advantage of known labels and the estimated missing labels of training instances, along with unlabeled instances to train a graph regularized semi-supervised multi-label linear classifier. Experimental results on public available multi-label datasets demonstrate that SMILE outperforms other related methods with respect to various multi-label learning evaluation metrics.

The reminder of this paper is organized as follows. Section 2 briefly reviews related work on multi-label learning with weak label. Section 3 describes the proposed approach. The experimental protocols and results are presented in Section 4. Conclusions are provided in Section 5.

2. Related work

In this section, we introduce some representative and related multi-label weak-label learning methods. These methods can be roughly categorized into two kinds, based on whether they employing unlabeled instances or not.

Supervised strategy assumes that sufficient labeled instances are available. For example, Bucak et al. [18] proposed a weak-label learning approach called MLR-GL. MLR-GL optimizes the ranking loss and group lasso in a convex optimization form, and it targets at predicting the labels of completely unlabeled new instances. Yu et al. [26] proposed a large-scale weak-label learning approach, which introduces a generic empirical risk minimization framework and solves the framework by exploiting the structure of squared loss function. Wu et al. [27] proposed an inductive multi-label missing label approach called MLML. MLML enforces the consistency between the predicted labels and the available labels, as well as the local smoothness among labels. Sun et al. [21] proposed a weak label learning approach based on three assumptions: (i) the decision boundary for each label should go across low density regions; (ii) each label generally has a much smaller number of positive examples than negative ones; (iii) there exists a group of low rank-based similarities and the similarity between instances can be approximated from these low rank-based ones. Based on these assumptions, they exploited convex optimization and quadratic programming to replenish missing labels of a partially labeled instance. Li et al. [28] first adopted a conditional restricted Boltzmann machine [29] to capture the high-order label dependence relationships in output space, and exploited label co-occurrence information retrieved from auxiliary resources [30] as prior knowledge. Then, they maximized the regularized marginal conditional likelihood of label vectors based on label dependence and co-occurrence information, to replenish the labels of partially labeled instances and to predict labels of completely unlabeled instances. These supervised methods limit their effectiveness by excluding a large amount of unlabeled instances, which can be used to boost the performance of multi-label classification [31].

Semi-supervised strategy exploits abundant unlabeled instances, along with some incompletely labeled instances, to replenish labels of partially labeled or completely unlabeled instances. For example, Wu et al. [24] proposed a semi-supervised weak-label learning method called SSW to replenish the labels of incompletely labeled instances. SSW firstly defines the positive, negative and missing labels of instances with entry values as 1, -1 and 0 of the initial label matrix. Then it constructs two graphs based on instances nodes and label nodes. Next, it combines these two graphs using a regularization framework constrained by label consistency and label smoothness, and solves the framework by pairwise constraint propagation [32] or Sylvester equation [33]. Wu et al. [25] proposed a mixed graph-based weak-label learning approach. This method encourages consistency between predicted and ground truth labels, encodes instance similarity and class label co-occurrence together as a convex quadratic matrix optimization problem, and then solves this problem by alternating direction method of multipliers [17]. Yu et al. [23] assumed the labels of instances depend on the feature information of instances and proposed an algorithm called ProDM. ProDM maximizes the dependency captured by Hilbert–Schmidt Independence Criterion [34] to replenish the missing labels of incompletely labeled instances and to predict labels of completely unlabeled instances. However, all these methods work under transductive setting and aim to make predictions for already known unlabeled instances. They can not be directly applied to new instances that are not available in the training process. Zhao and Guo [35] assumed the underlying label matrix is sparse and low rank, and simultaneously performed label matrix recovery within a low-rank sparse matrix recovery framework and semi-supervised multi-label learning with the manifold regularized [36] vector-valued model. However, for a new instance, this method has to compute its similarity with respect to all the training samples and then predicts the labels of that instance.

In this paper, we develop an inductive multi-label weak-label learning approach called SMILE to directly predict the labels of unlabeled instances using incomplete label information of training instances. SMILE first estimates the missing labels of an incompletely labeled instance based on the available labels of the instance and the empirical conditional probability between labels. It then takes advantages of labeled and unlabeled instances to construct a neighborhood graph. Next, SMILE exploits all the available labels (including estimated ones) of training instances to train a semi-supervised multi-label linear classifier regularized by a neighborhood graph. Different from [21,35], SMILE neither requires the label matrix being low-rank and sparse, nor stores all the training instances for follow-up prediction on new instances.

3. Problem and proposed solution

3.1. Problem formulation

In this section, we will introduce the measure for label correlation, estimation of missing labels and then propose the inductive semi-supervised multi-label classification model. Before that, we give some notations that will be used throughout this paper. Let $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N] \in \mathbb{R}^{D\times N}$ be a data matrix for N instances, $\mathbf{x}_i \in \mathbb{R}^D$ is the ith instance. $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_N] \in \mathbb{R}^{C\times N}$ is the initial label matrix for these N instances, where C is the number of distinct labels of these instances. $\mathbf{y}_i \in \mathbb{R}^C$ is the label vector of the ith instance. If instance i is annotated with label c, $\mathbf{y}_{ic} = 1$; otherwise, $\mathbf{y}_{ic} = 0$. Without losing generality, suppose among N instances in $\mathbf{X}$, the first l instances are partially labeled with some missing labels and the remaining u instances are unlabeled, $N = l + u$. Our goal is to use all the instances in $\mathbf{X}$ to train an inductive semi-supervised multi-label classifier to predict the labels of new unlabeled instances.

3.2. Measure label correlation

Unlike traditional single label learning methods, in multi-label learning, an instance usually has more than one label, and these labels usually interact with each other. It is important and necessary to exploit label correlation in multi-label learning [37–40]. Based on the order of correlations, existing label correlation exploitation strategies can be categorized as first-order, second-order and high-order ones [1,2]. Compared to first-order strategy which ignores label correlations, second-order strategy exploits label correlations to some extent. On the other hand, it leads to lower model and computational complexity than high-order strategy. In this paper, we employ the second-order strategy for its simplicity and effectiveness. The label correlation matrix $L \in \mathbb{R}^{C\times C}$ is defined as follows:

$$\mathbf{L}(c_1, c_2) = \frac{|\mathcal{Y}_{c_1} \bigcap \mathcal{Y}_{c_2}| + s}{|\mathcal{Y}_{c_1}| + 2s} \quad 1 \le c_1, c_2 \le C \tag{1}$$

where $\mathcal{Y}_{c_1}$ is the set of labeled instances annotated with c_1, $|\mathcal{Y}_{c_1}|$ is the number of labeled instances annotated with c_1. $|\mathcal{Y}_{c_1} \bigcap \mathcal{Y}_{c_2}|$ represents the number of labeled instances annotated with both c_1 and c_2, $s > 0$ is a smoothness parameter. The motivation to use s is for label imbalance. For example, suppose there are 30 articles, 5 of them are labeled with football, basketball and tennis, while the others are tagged with football and basketball. If we randomly pick up 10 articles to estimate the label correlation and these 10 articles are labeled with football and basketball only, then the correlation between football and tennis and that between basketball and sport are estimated as zeros. However, in fact, these three labels are correlated with each other. By setting $s > 0$, we can avoid these extreme cases, caused by label imbalance, to some extent.

3.3. Estimate missing labels

Based on the definition of label correlation L in Eq. (1), we can estimate the likelihood of a missing label c for the (ith $1 \le i \le l$) instance as follows:

$$\tilde{\mathbf{y}}_{ic} = \begin{cases} \mathbf{y}_i^T L(\cdot, c), & \text{if } \mathbf{y}_{ic} = 0 \\ 1, & \text{otherwise} \end{cases} \tag{2}$$

To ensure $\tilde{\mathbf{y}}_{ic} \in [0, 1]$, $\tilde{\mathbf{y}}_{ic}$ is normalized as $\tilde{\mathbf{y}}_{ic}/||\tilde{\mathbf{y}}_i||$ for $\mathbf{y}_{ic} = 0$. Eq. (2) is to replenish the missing label of an incompletely annotated instance using the already known labels of the instance and the label correlation. For example, if $\mathbf{y}_{ic} = 0$ but c has large correlations with the labels already annotated to the ith instance, then c may be a missing label for that instance and $\tilde{\mathbf{y}}_{ic}$ will be assigned with a large value. In other words, if we see gulls in a picture, then we may also find sea or islands in that picture, but seldom see tigers and grassland. $\tilde{\mathbf{y}}_i$ is the label vector for the ith instance with both known and estimated labels.

3.4. Semi-supervised multi-label linear classifier

To this end, stimulated by the work of Yu et al. [41], we extend their basic linear classifier to multi-label classification, and introduce a graph-based semi-supervised linear classifier to leverage labeled and unlabeled instances for multi-label weak-label learning. At first, we construct a k nearest neighbor (kNN) graph, where each node represents an instance, the weighted edge between two nodes represents the similarity between them. In this way, we can leverage the information of unlabeled instances and labeled instances to train a semi-supervised classifier. The weighted adjacent matrix $\mathbf{W} \in \mathbb{R}^{N\times N}$ of the kNN graph is specified as:

$$\mathbf{w}_{ij} = \begin{cases} 1, & \text{if } \mathbf{x}_i \in kNN(\mathbf{x}_j) \text{ or } \mathbf{x}_j \in kNN(\mathbf{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

where $\mathbf{w}_{ij}$ encodes the weight of edge between instances i and j, $\mathbf{x}_i \in kNN(\mathbf{x}_j)$ stands for $\mathbf{x}_i$ is one of the k nearest neighbors of $\mathbf{x}_j$, and the neighborhood relationship between instances is determined by Euclidean distance. We just use 0–1 weight in Eq. (3) for it simplicity and wide application. Other specifications of $\mathbf{W}$ can also be used.

Here, we train a graph-based semi-supervised linear classifier by minimizing the following objective function:

$$\Psi(f) = \Omega(f) + \alpha||f||_l^2 \tag{4}$$

Our motivation to adopt this objective function is based on two assumptions [42]: consistency assumption and smoothness assumption. The first term on the right side of Eq. (4) is based on consistency assumption, it measures the empirical loss on labeled instances and ensures the predicted label matrix be consistent with the initial label matrix. The second term is the regularization on labeled and unlabeled instances and is based on smoothness assumption, it ensures similar instances having similar predicted outputs. $\alpha \ge 0$ balances the importance of two terms.

The general form of a linear classifier $f(\mathbf{x})$ can be defined as:

$$f(\mathbf{x}) = \mathbf{P}^T\mathbf{x} + \mathbf{b} \tag{5}$$

where $\mathbf{P} \in \mathbb{R}^{D\times C}$ is the predictive matrix, $\mathbf{b} \in \mathbb{R}^C$ is the label bias, $f(\mathbf{x}) \in \mathbb{R}^C$ is the predicted likelihood vector for $\mathbf{x}$ with respect to C different labels.

The first part of Eq. (4) can be computed as follow:

$$\begin{aligned}\Omega(f) &= \sum_{i=1}^{l} ||\mathbf{P}^T\mathbf{x}_i + \mathbf{b} - \tilde{\mathbf{y}}_i||^2 \\ &= \sum_{i=1}^{N} (\mathbf{P}^T\mathbf{x}_i + \mathbf{b} - \tilde{\mathbf{y}}_i)^T \mathbf{h}_{ii} (\mathbf{P}^T\mathbf{x}_i + \mathbf{b} - \tilde{\mathbf{y}}_i) \\ &= tr((\mathbf{P}^T\mathbf{X} + \mathbf{b}\mathbf{1}^T - \tilde{\mathbf{Y}})^T \mathbf{H} (\mathbf{P}^T\mathbf{X} + \mathbf{b}\mathbf{1}^T - \tilde{\mathbf{Y}})) \end{aligned} \quad (6)$$

where $tr()$ is the matrix trace operator, $\tilde{\mathbf{Y}} = [\tilde{\mathbf{y}}_1, \tilde{\mathbf{y}}_2, \ldots, \tilde{\mathbf{y}}_N]$, $\mathbf{1} \in \mathbb{R}^N$ with all elements are 1s. $\mathbf{H} \in \mathbb{R}^{N\times N}$ is a diagonal matrix specified as follow:

$$\mathbf{h}_{ii} = \begin{cases} 1, & \text{if } \mathbf{x}_i \text{ is labeled} \\ 0, & \text{otherwise} \end{cases} \quad (7)$$

The second part of Eq. (4) is defined as:

$$\begin{aligned}||f||_I^2 &= \frac{1}{2}\sum_{i,j=1}^{N} ||f(\mathbf{x}_i) - f(\mathbf{x}_j)||^2 \mathbf{w}_{ij} \\ &= \frac{1}{2}\sum_{i,j=1}^{N} ||\mathbf{P}^T\mathbf{x}_i - \mathbf{P}^T\mathbf{x}_j||^2 \mathbf{w}_{ij} \\ &= tr(\mathbf{P}^T \sum_{i=1}^{N} (\mathbf{x}_i \mathbf{w}_{ii} \mathbf{x}_i^T)\mathbf{P} - \mathbf{P}^T (\sum_{i,j=1}^{N} (\mathbf{x}_i \mathbf{w}_{ij} \mathbf{x}_j^T)\mathbf{P}) \\ &= tr(\mathbf{P}^T\mathbf{X}(\mathbf{\Lambda} - \mathbf{W})\mathbf{X}^T\mathbf{P}) \\ &= tr(\mathbf{P}^T\mathbf{X}\mathbf{M}\mathbf{X}^T\mathbf{P}) \end{aligned} \quad (8)$$

where $\mathbf{\Lambda}$ is a diagonal matrix with $\Lambda_{ii} = \sum_{j=1}^{N} \mathbf{w}_{ij}$, $\mathbf{M} = \mathbf{\Lambda} - \mathbf{W}$ is the graph Laplacian matrix [43]. Based on Eqs. (6) and (8), we can rewrite Eq. (4) as:

$$\begin{aligned}\Psi(\mathbf{X}, \mathbf{P}, \mathbf{b}) = &tr((\mathbf{P}^T\mathbf{X} + \mathbf{b}\mathbf{1}^T - \tilde{\mathbf{Y}}^T)\mathbf{H}(\mathbf{P}^T\mathbf{X} + \mathbf{b}\mathbf{1}^T - \tilde{\mathbf{Y}}^T)) \\ &+\alpha tr(\mathbf{P}^T\mathbf{X}\mathbf{M}\mathbf{X}^T\mathbf{P}) \end{aligned} \quad (9)$$

The reason to minimize the first term of Eq. (9) is to force the predicted outputs $f(\mathbf{x})$ be similar to the original labels and estimated labels. The motivation to minimize the second term is to ensure similar instances having similar outputs. This motivation is often regarded as smoothness assumption [44]. In other words, if two instances $\mathbf{x}_i$ and $\mathbf{x}_j$ are quite similar to each other, then $\mathbf{w}_{ij}$ has a high value, $f(\mathbf{x}_i)$ and $f(\mathbf{x}_j)$ should have similar outputs; otherwise, there is a heavy penalty between them and causing a big loss. In this way, if $\mathbf{x}_j$ is a unlabeled instance and it is among k nearest neighbors of $\mathbf{x}_i$ whose labels are known, then $\mathbf{x}_j$ may be annotated with some labels of $\mathbf{x}_i$. In addition, the second term can also replenish the missing labels of incompletely labeled instances based on this assumption. Suppose $\mathbf{x}_i$ and $\mathbf{x}_j$ are reciprocal neighbors and $\mathbf{w}_{ij} = 1$. If $\mathbf{y}_{ic_1} = 1$, and $\mathbf{y}_{jc_2} = 1$, then it is quite likely that $\mathbf{x}_i$ has a missing label c_2 and $\mathbf{x}_j$ has a missing label c_1. Minimizing the second term of Eq. (9) can also help to replenish these missing labels of respective instances.

To obtain $\mathbf{P}$ and $\mathbf{b}$, we take partial derivative of $\Psi(\mathbf{X}, \mathbf{P}, \mathbf{b})$ with respect to $\mathbf{P}$ and $\mathbf{b}$ in Eq. (9) as follow:

$$\frac{\partial \Psi}{\partial \mathbf{P}} = 2\mathbf{X}\mathbf{H}(\mathbf{X}^T\mathbf{P} + \mathbf{1}\mathbf{b}^T - \tilde{\mathbf{Y}}) + 2\alpha\mathbf{X}\mathbf{M}\mathbf{X}^T\mathbf{P} \quad (10)$$

$$\frac{\partial \Psi}{\partial \mathbf{b}} = 2(\mathbf{P}^T\mathbf{X} + \mathbf{b}\mathbf{1}^T - \tilde{\mathbf{Y}}^T)\mathbf{H}\mathbf{1} \quad (11)$$

Let $\frac{\partial \Psi}{\partial \mathbf{P}} = 0$ and $\frac{\partial \Psi}{\partial \mathbf{b}} = 0$, we get the final solutions of $\mathbf{P}$ and $\mathbf{b}$ as:

$$\mathbf{P} = (\mathbf{X}\mathbf{H}_c\mathbf{X}^T + \alpha\mathbf{X}\mathbf{M}\mathbf{X}^T)^{-1}\mathbf{X}\mathbf{H}_c\tilde{\mathbf{Y}} \quad (12)$$

$$\mathbf{b} = \frac{(\tilde{\mathbf{Y}}^T - \mathbf{P}^T\mathbf{X})\mathbf{H}\mathbf{1}}{N} \quad (13)$$

where $\mathbf{H}_c$ is:

$$\mathbf{H}_c = \mathbf{H} - \frac{\mathbf{H}\mathbf{1}\mathbf{1}^T\mathbf{H}^T}{N} \quad (14)$$

The main procedure of SMILE is described in Algorithm 1.

Algorithm 1 SMILE: Semi-supervised multi-label classification using incomplete label information.

Input:

$\mathbf{X}$: Training samples, $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_N]$
$\mathbf{Y}$: Initial label matrix for N instances
k: The number of nearest neighbors used in Eq. (3)
α: Parameter used in Eq. (4)
$\mathbf{x}$: New unlabeled instance

Output:

$f(\mathbf{x})$: Predicted label likelihood vector for $\mathbf{x}$

Training:

1: Calculate the correlation matrix $\mathbf{L}$ using Eq. (1)
2: Estimate missing labels using Eq. (2)
3: Define the weighted adjacent matrix $\mathbf{W}$ using Eq. (3)
4: Construct the graph Laplacian matrix $\mathbf{M}$
5: Solve $\mathbf{P}$ and $\mathbf{b}$ using Eqs. (12)–(14)
6: **Return** $f(\mathbf{x}) = \mathbf{P}^T\mathbf{x} + \mathbf{b}$ by using Eq. (5)

3.5. Time complexity analysis

The time complexity of constructing kNN graph and label correlation matrix $\mathbf{L}$ are $O(N^2D)$ and $O(N^2C)$. In practice, the most time-consuming step of SMILE is to solve $\mathbf{P}$. To solve $\mathbf{P}$, we only need to solve a linear system of equations $(\mathbf{X}\mathbf{M}_1\mathbf{X}^T)\mathbf{P} = \mathbf{X}\mathbf{H}_c\tilde{\mathbf{Y}}$, where $\mathbf{M}_1 = (\mathbf{H}_c + \alpha\mathbf{M})$ and $\mathbf{M}_1 \in \mathbb{R}^{N\times N}$. The time complexity is dominated by matrix multiplication and matrix inverse. The complexity of matrix multiplication is $O(ND^2 + NDC)$ and the complexity of matrix inverse is $O(D^3)$, so the time complexity of SMILE is $O(N^2C + N^2D + ND^2 + D^3)$. For large scale datasets, the nearest neighbors can be efficiently determined by advanced techniques (i.e., tree-based data structures [45]) and matrix multiplication can be efficiently computed, since $\mathbf{M}_1$ is a sparse matrix with about $(k+1)N$ nonzero entries. To avoid large matrix inverse, iterative approximation algorithms (i.e., Conjugate Gradient (CG)[46]) can be applied. CG performs cheap updates and offers a good approximate solution within a few iterations, it is guaranteed to terminate in N steps and its computational complexity is proportional to the number of nonzero elements in $\mathbf{X}\mathbf{M}_1\mathbf{X}^T$. CG often terminates in few iterations. For this reason, the real runtime cost of the above operations can be greatly reduced. In fact, our following runtime cost comparison with other comparing methods on multi-label datasets shows that SMILE generally runs much faster than other comparing methods.

4. Experiments

4.1. Experimental setup

4.1.1. Datasets

We conduct experiments on three multi-label datasets, Cal500 [47], Bibtex [48] and Delicious [49], to investigate the performance of SMILE. Cal500 is a music multi-label dataset with 502 instances in 174 object classes. Each music is represented as a vector with 68 features. Bibtex and Delicious are large text datasets, similar to iMLCU [15], we keep the top 20% frequent words and filter rare categories by keeping top 30% frequent categories. In the end, we get 368 words and 48 topics for Bibtex, and 295 words and 100 topics for Delicious. The statistical information of these datasets are revealed in Table 1.

Table 1
Statistic of the experimental datasets. N is the number of instances, D is the dimensionality of instances, C is the number of distinct labels of instances, Avg. is the average number of labels per instance.

Datasets	N	D	C	Avg.
Cal500	502	68	174	26.044
Bibtex	7395	368	48	1.322
Delicious	16,105	295	100	15.323

In order to simulate the scenario of missing labels, we assume that currently available labels of a labeled instance are complete and randomly mask some labels of the instance. These masked labels are considered missing for that instance. For representation, we use m to record the number of masked labels. For example, if instance i has 5 labels, $m = 2$ means two labels are masked, namely two 1 s of $\mathbf{y}_i$ are changed to two 0 s. If an instance has fewer than m labels, we do not mask all the labels and ensure it have one label.

4.1.2. Comparing methods

To comparatively study the performance of SMILE, we compare it with six representative and related methods, ML-LOC [8], MLR-GL [18], MLML [27], Tram [11], SSW [24] and ProDM [23]. The first three are supervised multi-label classifiers, and the latter three are transductive multi-label classifiers. ML-LOC and Tram assume the training instances are completely labeled, while MLR-GL, MLML, SSW and ProDM assume training instances are incompletely labeled. To further study the effectiveness of SMILE in utilizing unlabeled instances and in employing label correlation, we introduce three variants of SMILE: (i) SMILE-Nc is adopted from SMILE without using the label correlation; (ii) SMILE-Nu is adopted from SMILE by excluding unlabeled instances, namely $\alpha = 0$; and (iii) SMILE-Ncu is adapted from SMILE by excluding unlabeled instances and label correlation.

Tram, SSW and ProDM are transductive classifiers that can not directly apply to new instances, which are unseen in the training process. Follow the suggestion in [9], we extend them for new instances by setting the labels of a new instance as the labels of its nearest training instance, and the labels of unlabeled training instances are predicted by the respective transductive classifier in advance. We want to remark that SMILE and SMILE-Nc directly use all the labeled and unlabeled instances to predict the labels of a new instance, without predicting the labels of unlabeled training instances in advance. ML-LOC, MLML, MLR-GL, SMILE-Nu and SMILE-Ncu directly use the labeled instances to predict the labels of a new instance without using the unlabeled instances. In the preliminary experiments, five-fold cross validation is conducted on Cal500, Bibtex and Delicious datasets to tune the parameters of ML-LOC, MLR-GL, MLML, Tram, SSW and ProDM in the ranges as the author suggested or in the same range as that of SMILE. In our experiments, parameters α, s and k for SMILE are tuned by cross validation within the range [0.01,1], [0.01,1] and [3,20], respectively. Experimental results show that SMILE yields relatively stable performance with α around 0.35, s around 0.5 and k around 5. SMILE sets α as 0.35, s as 0.5 and k as 5 for experiments.

4.1.3. Evaluation metrics

Multi-label classification can be evaluated by various metrics, each metric captures a specific aspect of the classifier [1,2]. Here we use four metrics: *RankingLoss, Coverage, Average Precision(AvgPrec)* and adapted *AUC* [18]. The first three metrics can be found in reference [1] and [2]. The adaptive *AUC* is suggested in [18]. *RankingLoss* counts the average number of times that irrelevant labels are ranked higher than relevant labels. *Coverage* evaluates how far we need, on average, to go down the ranked list of labels of $f(\mathbf{x})$ in order to cover all the relevant labels of $\mathbf{x}$. *AvgPrec* evaluates the average fraction of relevant labels ranked higher than a particular label. Adapted *AUC* is initially used by MLR-GL [18]. It first ranks all the labels for each test instance in descending order of predicted likelihood scores. It varies the number of predicted labels from 1 to C and plots the receiver operator curve by calculating true positive rate and false positive rate. Next, it computes the area under curve and utilizes the score of this area to evaluate the results of multi-label classification. In addition, to get a global measure on predicting new labels, we also adopt another metric *Accuracy* computed as follows:

$$Accuracy = \frac{|\mathcal{Y} \cap \mathcal{F}_p|}{|\mathcal{Y}|} \tag{15}$$

where $\mathcal{F}_p$ is the predicted label set of testing instances, $\mathcal{Y}$ is the ground truth label set, and $|\mathcal{Y} \cap \mathcal{F}_p|$ counts how many labels are correctly predicted. $f(\mathbf{x}) \in \mathbb{R}^C$ in Eq. (4) is the predicted likelihood vector for instance $\mathbf{x}$. *Accuracy* requires the vector to be a binary indicator vector. Here, we consider the labels corresponding to the r largest entries of $f(\mathbf{x})$ as the predicted labels of the ith instance, where r is determined as the average number of labels (round to next integer) of labeled instances. From Table 1, r for Cal500 is 27 and for Bibtex is 2.

To maintain consistency with other evaluation metrics, we report $1 - RankLoss$ instead of *RankingLoss*. Thus, similar to other metrics (except *Coverage*), the higher the value of *1-RankLoss*, the better the performance is. These metrics evaluate multi-label classification from different aspects, it is difficult for a method performing better than other methods across all these metrics.

4.2. Experiment results

In this section, we conduct experiments to investigate the performance of SMILE in predicting the labels of unlabeled instances using partially labeled ones, and compare the performance of SMILE with that of other comparing methods. We randomly partition the instances of each dataset into two parts. The first part accounts for 70% instances used as the training set, the second part accounts for 30% instances used as the testing set. Here, we consider two different label ratios: 10% and 30%. 10% means we randomly select 10% instances from the training set as labeled instances and take the other training instances as unlabeled instances. In each random partition, we randomly mask $m = 1$ (or 2, 3) labels of each labeled instance and then make use of these incompletely labeled and completely unlabeled training instances to predict the labels of instances in the testing set. To reduce random effect, we repeat the individual partition and evaluation 10 times for each particular setting of m and of label ratio. The recorded results (average and standard deviation) are reported in Tables 2–4 for label ratio 10% and Tables 5–7 for label ratio 30%. In these tables, the best (or comparable best) results in each group of columns are in **boldface** with statistical significance examined via pairwise t-test at 95% significance level. ML-LOC utilizes a clustering algorithm for each label to generate the LOC code, however since the labeled instances are so scarce that the clustering algorithm fails sometimes. Therefore, we only record the results with successfully generated LOC and report the average results of these successful runs. If the clustering algorithm is always failed in these 10 independent runs, the results of ML-LOC will be not reported.

From the results reported in these tables, we can observe that the performance of all methods increases with the label ratio rising, and SMILE outperforms other competitive methods across all the evaluation metrics in most cases. In summary, out of 90 configurations (3 datasets $\times$ 2 kinds of label ratio $\times$ 5 evaluation metics $\times$ 3 settings of m), SMILE outperforms ML-LOC, MLR-GL, MLML,

Table 2
Experimental results of each multi-label learning algorithm (mean± std) on ***Delicious*** dataset with 10% labeled instances.

m	Algorithm	Accuracy	1-RankLoss	AvgPrec	AUC	Coverage↓
1	ML-LOC	0.267 ± 0.005	0.733 ± 0.002	0.257 ± 0.003	0.733 ± 0.002	229.874 ± 0.134
	MLR-GL	0.140 ± 0.005	0.673 ± 0.011	0.069 ± 0.002	0.689 ± 0.011	266.386 ± 2.513
	MLML	0.260 ± 0.010	0.749 ± 0.010	0.252 ± 0.047	0.760 ± 0.009	225.934 ± 3.002
	Tram	0.271 ± 0.002	0.570 ± 0.000	0.218 ± 0.002	0.785 ± 0.000	229.592 ± 0.075
	ProDM	0.250 ± 0.002	0.735 ± 0.002	0.234 ± 0.001	0.757 ± 0.002	223.443 ± 0.338
	SSW	0.277 ± 0.001	0.747 ± 0.003	0.211 ± 0.001	0.778 ± 0.000	252.607 ± 0.116
	SMILE	**0.321 ± 0.001**	**0.801 ± 0.001**	**0.326 ± 0.001**	**0.805 ± 0.000**	**201.890 ± 0.352**
2	ML-LOC	0.261 ± 0.000	0.725 ± 0.000	0.253 ± 0.001	0.726 ± 0.001	229.570 ± 0.475
	MLR-GL	0.142 ± 0.002	0.671 ± 0.009	0.070 ± 0.001	0.677 ± 0.008	267.328 ± 2.263
	MLML	0.255 ± 0.001	0.746 ± 0.001	0.236 ± 0.006	0.756 ± 0.002	225.379 ± 1.103
	Tram	0.271 ± 0.001	0.571 ± 0.002	0.220 ± 0.002	0.783 ± 0.001	230.984 ± 0.730
	ProDM	0.246 ± 0.001	0.729 ± 0.004	0.230 ± 0.000	0.753 ± 0.002	225.755 ± 0.063
	SSW	0.272 ± 0.001	0.744 ± 0.003	0.210 ± 0.001	0.773 ± 0.001	252.880 ± 0.350
	SMILE	**0.319 ± 0.001**	**0.801 ± 0.001**	**0.323 ± 0.001**	**0.805 ± 0.001**	**201.488 ± 0.514**
3	ML-LOC	0.261 ± 0.000	0.725 ± 0.000	0.253 ± 0.001	0.726 ± 0.001	229.570 ± 0.475
	MLR-GL	0.142 ± 0.002	0.671 ± 0.009	0.070 ± 0.001	0.677 ± 0.008	267.328 ± 2.263
	MLML	0.246 ± 0.002	0.735 ± 0.001	0.237 ± 0.004	0.746 ± 0.003	230.180 ± 2.478
	Tram	0.271 ± 0.001	0.571 ± 0.002	0.220 ± 0.002	0.783 ± 0.001	230.984 ± 0.730
	ProDM	0.246 ± 0.001	0.729 ± 0.004	0.230 ± 0.000	0.753 ± 0.002	225.755 ± 0.063
	SSW	0.270 ± 0.001	0.739 ± 0.001	0.208 ± 0.000	0.769 ± 0.000	253.112 ± 0.473
	SMILE	**0.319 ± 0.001**	**0.801 ± 0.001**	**0.323 ± 0.001**	**0.805 ± 0.001**	**201.488 ± 0.514**

Table 3
Experimental results of each multi-label learning algorithm (mean± std) on ***Cal500*** dataset with 10% labeled instances.

m	Algorithm	Accuracy	1-RankLoss	AvgPrec	AUC	Coverage↓
1	ML-LOC	0.441 ± 0.004	0.699 ± 0.002	0.483 ± 0.003	0.696 ± 0.002	93.170 ± 0.292
	MLR-GL	**0.454 ± 0.003**	**0.724 ± 0.001**	**0.502 ± 0.003**	**0.721 ± 0.001**	91.312 ± 0.228
	MLML	0.442 ± 0.004	0.715 ± 0.003	0.492 ± 0.004	0.712 ± 0.003	**90.780 ± 0.454**
	Tram	0.391 ± 0.003	0.644 ± 0.003	0.411 ± 0.003	0.687 ± 0.002	**90.826 ± 0.271**
	ProDM	0.444 ± 0.004	0.718 ± 0.002	0.501 ± 0.003	0.715 ± 0.002	**90.701 ± 0.445**
	SSW	0.431 ± 0.007	0.708 ± 0.010	0.482 ± 0.006	0.704 ± 0.003	**90.821 ± 0.266**
	SMILE	**0.452 ± 0.003**	**0.720 ± 0.002**	**0.503 ± 0.003**	0.717 ± 0.002	**90.692 ± 0.270**
2	ML-LOC	0.422 ± 0.004	0.673 ± 0.003	0.454 ± 0.003	0.674 ± 0.003	94.252 ± 0.244
	MLR-GL	**0.451 ± 0.002**	**0.722 ± 0.001**	**0.503 ± 0.003**	**0.719 ± 0.001**	91.074 ± 0.410
	MLML	0.442 ± 0.007	0.714 ± 0.004	0.490 ± 0.006	0.711 ± 0.004	91.100 ± 0.142
	Tram	0.388 ± 0.006	0.649 ± 0.004	0.409 ± 0.005	0.686 ± 0.003	90.470 ± 0.306
	ProDM	0.441 ± 0.004	0.710 ± 0.003	0.488 ± 0.005	0.708 ± 0.003	91.291 ± 0.277
	SSW	0.432 ± 0.001	0.712 ± 0.007	0.491 ± 0.005	0.709 ± 0.004	**89.666 ± 0.333**
	SMILE	**0.452 ± 0.003**	**0.724 ± 0.003**	**0.504 ± 0.004**	**0.722 ± 0.003**	90.838 ± 0.330
3	ML-LOC	0.421 ± 0.004	0.673 ± 0.004	0.463 ± 0.004	0.672 ± 0.003	94.477 ± 0.269
	MLR-GL	0.442 ± 0.003	0.715 ± 0.002	0.491 ± 0.003	0.713 ± 0.002	**91.105 ± 0.228**
	MLML	0.447 ± 0.006	0.716 ± 0.005	**0.500 ± 0.007**	0.712 ± 0.005	**90.359 ± 0.329**
	Tram	0.390 ± 0.004	0.646 ± 0.005	0.411 ± 0.004	0.682 ± 0.002	91.852 ± 0.256
	ProDM	0.441 ± 0.003	0.709 ± 0.003	0.489 ± 0.004	0.707 ± 0.003	91.663 ± 0.207
	SSW	0.431 ± 0.006	0.709 ± 0.009	0.486 ± 0.006	0.705 ± 0.003	**90.591 ± 0.384**
	SMILE	**0.447 ± 0.001**	**0.722 ± 0.003**	**0.503 ± 0.003**	**0.719 ± 0.003**	**90.793 ± 0.377**

Tram, ProDM and SSW of 87.78%, 91.11%, 94.44%, 96.67%, 98.89% and 85.56%, ties with them of 11.11%, 7.78%, 5.56%, 3.33%, 1.11% and 7.78% cases, and loses to MLLOC, MLR-GL and SSW in 1.11%, 1.11% and 6.67% cases, respectively. Taking *Accuracy* for example, SMILE on average improves ML-LOC, MLR-GL, MLML, Tram, ProDM and SSW by 77.78%, 94.44%, 94.44%, 88.89%, 100.00% and 83.33%. These results corroborate the effectiveness of SMILE on predicting labels of multi-label instances using incomplete label information.

Taking experimental results on *Delicious* dataset in Table 2 for example, we can observe that SMILE achieves the best (or comparable best) performance among all the comparing methods across these evaluation metrics. Both SMILE and Tram exploit unlabeled instances, but Tram is outperformed by SMILE. The possible reason is that Tram targets at predicting unlabeled training instances under the assumption that the labels of annotated instances are complete. Here it is adapted for predicting new unlabeled instances using partially labeled instances. ProDM optimizes an objective function based on the label correlation, smoothness assumption between neighborhood instances and the dependency between the labels and features of instances. In fact, SMILE also employs the label correlation and smoothness assumption, it always outperforms ProDM. That is because ProDM is a transductive approach and targets at predicting the labels of unlabeled training instances. SSW assumes missing labels to be intermediate values (0) between negative (−1) and positive (1) labels, and predicts labels for unlabeled instances by enforcing consistency with available labels and smoothness between labels. Both SMILE and SSW exploit the label correlation and smoothness assumption, however, SSW is outperformed by SMILE in many cases. There are two possible reasons: (i) SMILE does not give any bias to missing labels, while SSW sets missing labels to be intermediate value (0) and brings in label bias, since negative labels are generally unknown in advance; (ii) SSW is a transductive method and aims to predict the labels of unlabeled instances in training set, whereas SMILE is an inductive method and targets at predicting the labels of unlabeled instances not included in the training set.

Table 4
Experimental results of each multi-label learning algorithm (mean± std) on ***Bibtex*** dataset with 10% labeled instances.

m	Algorithm	Accuracy	1-RankLoss	AvgPrec	AUC	Coverage↓
1	ML-LOC	**0.296 ± 0.002**	**0.791 ± 0.004**	**0.463 ± 0.002**	0.786 ± 0.004	**12.961 ± 0.231**
	MLR-GL	0.133 ± 0.001	0.642 ± 0.002	0.230 ± 0.001	0.621 ± 0.002	21.893 ± 0.097
	MLML	0.237 ± 0.004	0.687 ± 0.066	0.277 ± 0.008	0.716 ± 0.013	20.419 ± 1.055
	Tram	0.244 ± 0.002	0.777 ± 0.002	0.356 ± 0.005	0.779 ± 0.002	13.743 ± 0.091
	ProDM	0.133 ± 0.004	0.619 ± 0.003	0.225 ± 0.002	0.621 ± 0.002	22.683 ± 0.167
	SSW	0.262 ± 0.001	0.758 ± 0.001	0.362 ± 0.000	**0.795 ± 0.000**	13.491 ± 0.412
	SMILE	0.250 ± 0.005	**0.794 ± 0.002**	0.411 ± 0.006	0.789 ± 0.003	**13.143 ± 0.150**
2	ML-LOC	**0.302 ± 0.006**	**0.789 ± 0.005**	**0.470 ± 0.006**	0.782 ± 0.005	**13.323 ± 0.270**
	MLR-GL	0.135 ± 0.001	0.647 ± 0.003	0.231 ± 0.003	0.623 ± 0.003	21.693 ± 0.193
	MLML	0.232 ± 0.005	0.685 ± 0.061	0.276 ± 0.002	0.719 ± 0.009	20.123 ± 1.715
	Tram	0.249 ± 0.003	0.772 ± 0.003	0.368 ± 0.004	0.771 ± 0.002	14.097 ± 0.150
	ProDM	0.133 ± 0.001	0.640 ± 0.003	0.228 ± 0.002	0.625 ± 0.002	21.869 ± 0.176
	SSW	0.261 ± 0.001	0.754 ± 0.003	0.359 ± 0.001	**0.793 ± 0.001**	**13.721 ± 0.350**
	SMILE	0.247 ± 0.004	**0.790 ± 0.002**	0.411 ± 0.003	0.781 ± 0.002	**13.517 ± 0.147**
3	ML-LOC	**0.295 ± 0.004**	0.797 ± 0.005	**0.466 ± 0.005**	0.778 ± 0.004	**13.304 ± 0.294**
	MLR-GL	0.135 ± 0.001	0.648 ± 0.002	0.232 ± 0.001	0.622 ± 0.003	21.588 ± 0.164
	MLML	0.239 ± 0.009	0.688 ± 0.066	0.296 ± 0.025	0.723 ± 0.013	19.054 ± 1.069
	Tram	0.248 ± 0.003	0.769 ± 0.003	0.358 ± 0.006	0.767 ± 0.003	14.468 ± 0.176
	ProDM	0.135 ± 0.001	0.646 ± 0.002	0.229 ± 0.002	0.623 ± 0.002	21.769 ± 0.126
	SSW	0.259 ± 0.001	0.749 ± 0.000	0.357 ± 0.000	**0.784 ± 0.001**	14.210 ± 0.362
	SMILE	0.237 ± 0.004	**0.786 ± 0.002**	0.401 ± 0.005	0.775 ± 0.002	**13.493 ± 0.179**

Table 5
Experimental results of each multi-label learning algorithm (mean± std) on ***Delicious*** dataset with 30% labeled instances.

m	Algorithm	Accuracy	1-RankLoss	AvgPrec	AUC	Coverage↓
1	ML-LOC	0.293 ± 0.001	0.742 ± 0.001	0.289 ± 0.001	0.749 ± 0.000	229.390 ± 1.083
	MLR-GL	0.144 ± 0.003	0.684 ± 0.011	0.071 ± 0.002	0.696 ± 0.011	262.847 ± 1.156
	MLML	0.260 ± 0.002	0.747 ± 0.004	0.242 ± 0.004	0.748 ± 0.004	225.371 ± 2.599
	Tram	0.283 ± 0.001	0.604 ± 0.000	0.223 ± 0.001	0.792 ± 0.000	218.209 ± 0.137
	ProDM	0.258 ± 0.001	0.741 ± 0.001	0.246 ± 0.000	0.765 ± 0.000	220.777 ± 0.751
	SSW	0.295 ± 0.001	0.767 ± 0.001	0.252 ± 0.001	0.789 ± 0.000	248.597 ± 0.404
	SMILE	**0.332 ± 0.001**	**0.818 ± 0.000**	**0.342 ± 0.001**	**0.821 ± 0.000**	**188.717 ± 0.262**
2	ML-LOC	0.288 ± 0.002	0.741 ± 0.001	0.284 ± 0.001	0.751 ± 0.001	228.964 ± 1.011
	MLR-GL	0.144 ± 0.003	0.679 ± 0.014	0.070 ± 0.003	0.687 ± 0.014	261.604 ± 1.716
	MLML	0.255 ± 0.009	0.745 ± 0.008	0.243 ± 0.007	0.757 ± 0.006	224.994 ± 1.916
	Tram	0.281 ± 0.001	0.612 ± 0.003	0.226 ± 0.000	0.790 ± 0.001	217.851 ± 0.920
	ProDM	0.251 ± 0.005	0.739 ± 0.004	0.241 ± 0.004	0.764 ± 0.002	219.683 ± 0.277
	SSW	0.292 ± 0.000	0.769 ± 0.001	0.249 ± 0.000	0.783 ± 0.001	248.985 ± 0.268
	SMILE	**0.331 ± 0.001**	**0.817 ± 0.000**	**0.341 ± 0.001**	**0.822 ± 0.000**	**188.693 ± 0.242**
3	ML-LOC	0.288 ± 0.002	0.741 ± 0.001	0.284 ± 0.001	0.751 ± 0.001	228.964 ± 1.011
	MLR-GL	0.144 ± 0.003	0.679 ± 0.014	0.070 ± 0.003	0.687 ± 0.014	261.604 ± 1.716
	MLML	0.255 ± 0.009	0.745 ± 0.008	0.243 ± 0.007	0.757 ± 0.006	224.994 ± 1.916
	Tram	0.281 ± 0.001	0.612 ± 0.003	0.226 ± 0.000	0.790 ± 0.001	217.851 ± 0.920
	ProDM	0.251 ± 0.005	0.739 ± 0.004	0.241 ± 0.004	0.764 ± 0.002	219.683 ± 0.277
	SSW	0.290 ± 0.001	0.762 ± 0.001	0.231 ± 0.001	0.782 ± 0.001	247.072 ± 0.344
	SMILE	**0.331 ± 0.001**	**0.817 ± 0.000**	**0.341 ± 0.001**	**0.822 ± 0.000**	**188.693 ± 0.242**

To extend ProDM, SSW and Tram for inductive classification, we first predict the labels of unlabeled training instances. For an unlabeled testing instance, its labels are determined by the original (or predicted) labels of its nearest training instance. This fact suggests the inductive approach is more suitable for unseen instances than transductive one. We also observe that Tram outperforms SSW in most cases, one principal reason is that Tram exploits a dimensionality reduction technique [50] to extract features at first, and then applies transductive classification on the dimensionality reduced instances.

Both ML-LOC and SMILE are inductive approaches. They explicitly utilize label correlation, but SMILE still outperforms ML-LOC. That is principally because ML-LOC assumes the training instances are fully labeled and it only exploits labeled instances, discarding abundant unlabeled instances in the training process. MLR-GL is another inductive approach. It predicts labels of unlabeled instances using incompletely labeled instances, but it is still outperformed by SMILE. The possible reason is that MLR-GL does not take advantage of unlabeled instances. MLML is another inductive approach for multi-label classification with missing labels, but it loses to SMILE in most cases. This is because MLML works under supervised setting, it assumes sufficient labeled training instances are available and discards abundant unlabeled instances in training process. Both MLML and SSW are recently proposed methods for handling missing labels in multi-label learning. Particularly, MLML outperforms MLLOC, MLR-GL, Tram and ProDM of 48.89%, 77.78%, 46.67% and 71.11% cases, and SSW outperforms them of 58.89%, 76.67%, 71.11% and 66.67% cases, respectively. SSW generally achieves better performance than MLML. That is because SSW utilizes abundant unlabeled instances in training process, whereas SSW does not. These comparisons indicate unlabeled instances can be used to boost the performance of multi-label weak-label learning.

An interesting observation is that ML-LOC has a similar (or comparable better) performance with SMILE on *Bibtex* dataset (see Table 4). That is principally because the average number of Bibtex

Table 6
Experimental results of each multi-label learning algorithm (mean± std) on ***Cal500*** dataset with 30% labeled instances.

m	Algorithm	Accuracy	1-RankLoss	AvgPrec	AUC	Coverage↓
1	ML-LOC	0.451 ± 0.003	0.708 ± 0.003	0.498 ± 0.004	0.705 ± 0.003	92.584 ± 0.308
	MLR-GL	0.446 ± 0.004	0.729 ± 0.002	0.491 ± 0.005	0.726 ± 0.002	90.346 ± 0.220
	MLML	0.460 ± 0.004	0.729 ± 0.003	0.518 ± 0.004	0.729 ± 0.003	**89.889 ± 0.454**
	Tram	0.329 ± 0.003	0.522 ± 0.005	0.341 ± 0.002	0.657 ± 0.002	90.569 ± 0.264
	ProDM	0.458 ± 0.002	0.731 ± 0.001	0.513 ± 0.001	0.728 ± 0.001	90.662 ± 0.268
	SSW	0.449 ± 0.002	0.716 ± 0.007	0.497 ± 0.004	0.718 ± 0.001	**89.909 ± 0.463**
	SMILE	**0.466 ± 0.001**	**0.736 ± 0.002**	**0.520 ± 0.002**	**0.733 ± 0.002**	**89.922 ± 0.218**
2	ML-LOC	0.440 ± 0.001	0.694 ± 0.004	0.482 ± 0.003	0.692 ± 0.004	93.450 ± 0.299
	MLR-GL	0.455 ± 0.002	0.727 ± 0.002	0.489 ± 0.005	0.725 ± 0.002	90.485 ± 0.201
	MLML	0.449 ± 0.004	0.728 ± 0.004	0.507 ± 0.005	0.725 ± 0.004	90.264 ± 0.340
	Tram	0.325 ± 0.002	0.513 ± 0.004	0.332 ± 0.002	0.655 ± 0.001	**90.071 ± 0.174**
	ProDM	0.449 ± 0.003	0.724 ± 0.003	0.503 ± 0.003	0.722 ± 0.003	90.711 ± 0.217
	SSW	0.441 ± 0.004	0.719 ± 0.004	**0.517 ± 0.001**	0.718 ± 0.002	**89.803 ± 0.255**
	SMILE	**0.463 ± 0.002**	**0.735 ± 0.002**	**0.519 ± 0.002**	**0.732 ± 0.002**	90.419 ± 0.356
3	ML-LOC	0.441 ± 0.002	0.695 ± 0.002	0.485 ± 0.003	0.694 ± 0.003	94.257 ± 0.161
	MLR-GL	0.450 ± 0.002	0.728 ± 0.001	0.492 ± 0.005	0.726 ± 0.001	90.422 ± 0.253
	MLML	0.450 ± 0.004	0.729 ± 0.002	0.504 ± 0.002	0.725 ± 0.002	90.391 ± 0.333
	Tram	0.333 ± 0.003	0.535 ± 0.005	0.341 ± 0.002	0.660 ± 0.002	90.356 ± 0.260
	ProDM	0.443 ± 0.007	0.720 ± 0.006	0.498 ± 0.008	0.717 ± 0.006	90.995 ± 0.342
	SSW	0.444 ± 0.004	0.721 ± 0.011	0.502 ± 0.004	0.721 ± 0.003	**89.313 ± 0.329**
	SMILE	**0.462 ± 0.002**	**0.735 ± 0.002**	**0.518 ± 0.002**	**0.732 ± 0.002**	**89.749 ± 0.306**

Table 7
Experimental results of each multi-label learning algorithm (mean± std) on ***Bibtex*** dataset with 30% labeled instances.

m	Algorithm	Accuracy	1-RankLoss	AvgPrec	AUC	Coverage↓
1	ML-LOC	**0.325 ± 0.002**	0.824 ± 0.002	0.495 ± 0.002	0.815 ± 0.002	11.777 ± 0.107
	MLR-GL	0.136 ± 0.002	0.657 ± 0.003	0.241 ± 0.003	0.635 ± 0.002	21.519 ± 0.138
	MLML	0.269 ± 0.004	0.723 ± 0.055	0.322 ± 0.007	0.783 ± 0.009	15.228 ± 1.116
	Tram	0.254 ± 0.003	0.798 ± 0.002	0.312 ± 0.005	0.817 ± 0.002	11.907 ± 0.104
	ProDM	0.135 ± 0.001	0.640 ± 0.002	0.230 ± 0.002	0.641 ± 0.002	21.755 ± 0.090
	SSW	0.288 ± 0.001	0.783 ± 0.001	0.396 ± 0.000	0.822 ± 0.000	**8.920 ± 0.473**
	SMILE	**0.329 ± 0.004**	**0.861 ± 0.001**	**0.517 ± 0.004**	**0.856 ± 0.001**	9.396 ± 0.062
2	ML-LOC	0.312 ± 0.003	0.820 ± 0.003	0.487 ± 0.005	0.810 ± 0.003	11.988 ± 0.167
	MLR-GL	0.134 ± 0.001	0.659 ± 0.002	0.255 ± 0.005	0.630 ± 0.002	21.318 ± 0.112
	MLML	0.257 ± 0.002	0.717 ± 0.054	0.320 ± 0.002	0.759 ± 0.008	15.961 ± 0.933
	Tram	0.260 ± 0.003	0.801 ± 0.002	0.332 ± 0.004	0.814 ± 0.001	11.909 ± 0.112
	ProDM	0.135 ± 0.002	0.648 ± 0.003	0.233 ± 0.002	0.634 ± 0.002	21.647 ± 0.142
	SSW	0.287 ± 0.001	0.775 ± 0.002	0.388 ± 0.001	0.819 ± 0.000	**9.356 ± 0.562**
	SMILE	**0.317 ± 0.003**	**0.856 ± 0.002**	**0.506 ± 0.003**	**0.848 ± 0.002**	**9.728 ± 0.102**
3	ML-LOC	**0.312 ± 0.004**	0.813 ± 0.001	0.480 ± 0.003	0.801 ± 0.002	12.574 ± 0.106
	MLR-GL	0.136 ± 0.002	0.662 ± 0.001	0.260 ± 0.004	0.631 ± 0.002	20.893 ± 0.080
	MLML	0.255 ± 0.004	0.710 ± 0.053	0.313 ± 0.007	0.766 ± 0.009	16.473 ± 1.006
	Tram	0.256 ± 0.003	0.800 ± 0.001	0.320 ± 0.004	0.814 ± 0.002	11.959 ± 0.070
	ProDM	0.135 ± 0.001	0.656 ± 0.002	0.236 ± 0.001	0.632 ± 0.002	21.430 ± 0.122
	SSW	0.285 ± 0.001	0.770 ± 0.002	0.388 ± 0.001	0.816 ± 0.001	**9.763 ± 0.438**
	SMILE	**0.313 ± 0.004**	**0.855 ± 0.002**	**0.500 ± 0.003**	**0.845 ± 0.002**	**9.743 ± 0.120**

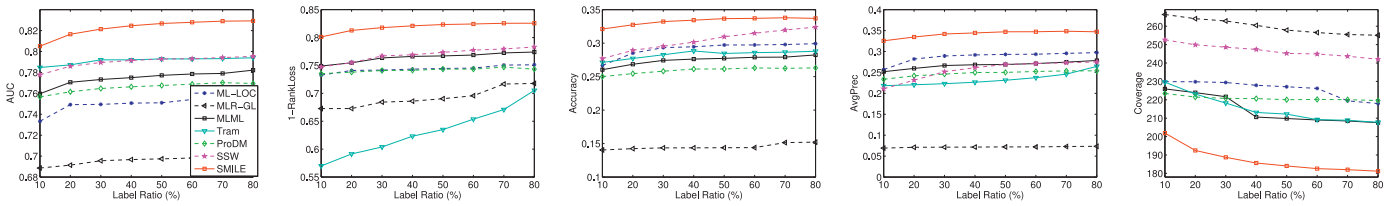
is 1.322 and the estimated label correlation matrix $\mathbf{L}$ is less reliable as more labels being masked.

To further investigate the performance trend of these comparing methods with more labeled instances, we conduct additional experiments on Delicious dataset to study the performance of SMILE with respect to different ratios of labeled training instances, and compare its performance with these comparing methods. Similar to previous experimental protocols, we randomly draw 70% instances from the Delicious as training set and the remaining 30% as testing set. Next, we increase the ratio of labeled instances from 10% to 80%, and keep the other instances in the training set as unlabeled instances. Ten independent experiments are conducted under each fixed ratio (from 10% to 80% with stepsize of 10%). Fig. 1 shows the performance of SMILE and other six related comparing methods with respect to AUC, 1-RankLoss, AvgPrec, Accuracy and Coverage under $m = 1$, respectively.

From Fig. 1, we can observe that the performance of SMILE and other six related methods increases with the increase of labeled instances and SMILE performs much better than other six comparing methods across five different evaluation metrics. Taking AUC in Fig. 1a for example, with ratio increasing from 10% to 80%, the AUC score of SMILE, ML-LOC, MLR-GL, MLML, Tram, ProDM and SSW increases by 2.42%, 2.22%, 1.51%, 2.26%, 0.93%, 1.28% and 1.76%, respectively. These observations show the superiority of SMILE in multi-label classification with missing labels than other six related methods, even with a large portion of labeled instances.

4.3. Components analysis

To investigate the benefit of using label correlation and abundant unlabeled instances, we conduct additional experiments with three variants of SMILE: SMILE-Nc, SMILE-Nu and SMILE-Ncu. Similarly to previous experimental protocols, we repeat the experiments 10 times for each fixed setting of m for a particular dataset. The experimental results with label ratio as 10% are reported in Tables 8–10. In these tables, the best (or comparable best) results



(a) AUC (b) 1-RankLoss (c) Accuracy (d) AvgPrec (e) Coverage

Fig. 1. Performance on **Delicious** dataset under different ratios of labeled instances.

Table 8
Experimental results of variants of SMILE (mean± std) on ***Delicious*** dataset.

m	Metric	SMILE-Nu	SMILE-Ncu	SMILE-Nc	SMILE
1	Accuracy	**0.320 ± 0.001**	**0.320 ± 0.001**	**0.321 ± 0.001**	**0.321 ± 0.001**
	1-RankLoss	0.782 ± 0.001	0.780 ± 0.001	0.781 ± 0.001	**0.801 ± 0.001**
	AvgPrec	**0.323 ± 0.001**	**0.322 ± 0.001**	**0.325 ± 0.001**	**0.326 ± 0.001**
	AUC	0.801 ± 0.000	0.798 ± 0.000	0.801 ± 0.001	**0.805 ± 0.000**
	Coverage↓	202.444 ± 0.390	204.898 ± 0.405	202.962 ± 0.373	**201.890 ± 0.352**
2	Accuracy	**0.319 ± 0.001**	**0.318 ± 0.001**	**0.317 ± 0.001**	**0.319 ± 0.001**
	1-RankLoss	0.780 ± 0.001	0.780 ± 0.001	0.780 ± 0.000	**0.801 ± 0.001**
	AvgPrec	**0.319 ± 0.001**	**0.318 ± 0.001**	**0.320 ± 0.001**	**0.323 ± 0.001**
	AUC	0.798 ± 0.000	0.780 ± 0.000	0.8000 ± 0.000	**0.805 ± 0.001**
	Coverage↓	204.191 ± 0.366	206.736 ± 0.375	204.594 ± 0.306	**201.488 ± 0.514**
3	Accuracy	**0.317 ± 0.001**	**0.317 ± 0.001**	**0.317 ± 0.000**	**0.316 ± 0.001**
	1-RankLoss	0.777 ± 0.001	0.770 ± 0.000	0.780 ± 0.001	**0.797 ± 0.000**
	AvgPrec	**0.316 ± 0.001**	**0.315 ± 0.001**	**0.319 ± 0.001**	**0.319 ± 0.001**
	AUC	0.780 ± 0.000	0.760 ± 0.000	0.788 ± 0.000	**0.802 ± 0.000**
	Coverage↓	206.595 ± 0.374	209.326 ± 0.382	206.141 ± 0.374	**204.410 ± 0.370**

Table 9
Experimental results of variants of SMILE (mean± std) on ***Cal500*** dataset.

m	Metric	SMILE-Nu	SMILE-Ncu	SMILE-Nc	SMILE
1	Accuracy	0.416 ± 0.004	0.409 ± 0.004	0.445 ± 0.003	**0.452 ± 0.003**
	1-RankLoss	0.678 ± 0.002	0.670 ± 0.003	0.706 ± 0.002	**0.720 ± 0.002**
	AvgPrec	0.453 ± 0.004	0.445 ± 0.004	0.495 ± 0.003	**0.503 ± 0.003**
	AUC	0.678 ± 0.002	0.671 ± 0.003	0.703 ± 0.002	**0.717 ± 0.002**
	Coverage↓	93.566 ± 0.174	93.899 ± 0.174	**90.765 ± 0.206**	**90.692 ± 0.270**
2	Accuracy	0.422 ± 0.004	0.414 ± 0.004	**0.452 ± 0.004**	**0.452 ± 0.003**
	1-RankLoss	0.682 ± 0.004	0.674 ± 0.004	0.703 ± 0.002	**0.724 ± 0.003**
	AvgPrec	0.464 ± 0.004	0.455 ± 0.004	**0.503 ± 0.003**	**0.504 ± 0.004**
	AUC	0.681 ± 0.004	0.673 ± 0.004	0.701 ± 0.002	**0.722 ± 0.003**
	Coverage↓	93.172 ± 0.373	93.577 ± 0.356	**90.909 ± 0.367**	**90.838 ± 0.330**
3	Accuracy	0.414 ± 0.004	0.408 ± 0.004	**0.441 ± 0.003**	**0.447 ± 0.005**
	1-RankLoss	0.678 ± 0.003	0.670 ± 0.003	0.712 ± 0.002	**0.722 ± 0.003**
	AvgPrec	0.454 ± 0.004	0.446 ± 0.004	**0.502 ± 0.002**	**0.503 ± 0.003**
	AUC	0.677 ± 0.004	0.669 ± 0.004	0.709 ± 0.002	**0.719 ± 0.003**
	Coverage↓	93.616 ± 0.226	93.984 ± 0.220	**91.094 ± 0.273**	**90.793 ± 0.377**

Table 10
Experimental results of variants of SMILE (mean± std) on ***Bibtex*** dataset.

m	Metric	SMILE-Nu	SMILE-Ncu	SMILE-Nc	SMILE
1	Accuracy	**0.284 ± 0.006**	0.253 ± 0.006	0.247 ± 0.004	0.250 ± 0.005
	1-RankLoss	0.699 ± 0.004	0.702 ± 0.004	**0.793 ± 0.002**	**0.794 ± 0.002**
	AvgPrec	0.307 ± 0.007	0.309 ± 0.007	**0.411 ± 0.004**	**0.411 ± 0.006**
	AUC	0.691 ± 0.003	0.696 ± 0.003	0.783 ± 0.001	**0.789 ± 0.003**
	Coverage↓	18.725 ± 0.130	18.479 ± 0.133	**13.030 ± 0.126**	**13.143 ± 0.150**
2	Accuracy	**0.289 ± 0.003**	0.249 ± 0.003	0.241 ± 0.004	0.247 ± 0.004
	1-RankLoss	0.692 ± 0.003	0.693 ± 0.003	**0.788 ± 0.001**	**0.790 ± 0.002**
	AvgPrec	0.315 ± 0.004	0.312 ± 0.004	**0.412 ± 0.003**	**0.411 ± 0.003**
	AUC	0.681 ± 0.003	0.682 ± 0.003	0.772 ± 0.003	**0.781 ± 0.002**
	Coverage↓	19.197 ± 0.163	19.123 ± 0.167	**13.346 ± 0.139**	**13.517 ± 0.147**
3	Accuracy	**0.290 ± 0.005**	0.243 ± 0.005	0.241 ± 0.002	0.237 ± 0.004
	1-RankLoss	0.696 ± 0.003	0.696 ± 0.003	0.773 ± 0.003	**0.786 ± 0.002**
	AvgPrec	0.317 ± 0.006	0.315 ± 0.006	**0.408 ± 0.005**	0.401 ± 0.005
	AUC	0.685 ± 0.003	0.685 ± 0.003	0.763 ± 0.003	**0.775 ± 0.002**
	Coverage↓	18.930 ± 0.149	18.959 ± 0.155	**13.124 ± 0.200**	**13.493 ± 0.179**

are in **boldface** and the statistical significance is examined via pairwise t-test at 95% significance level.

From these Tables, we can observe that simultaneously using label correlation and unlabeled instances can achieve the best results in most cases. In summary, out of 45 configurations (3 datasets × 5 evaluation metics × 3 settings of m), SMILE outperforms SMILE-Nc of 51.11% the cases, SMILE-Nu of 80.00% the cases, SMILE-Ncu of 84.44% the cases, ties with SMILE-Nc, SMILE-Nu and SMILE-Ncu of 48.89%, 13.33% and 8.89% the cases, and loses to SMILE-Nu and SMILE-Ncu of 6.67% and 6.67% the cases, respectively. SMILE-Nu improves SMILE-Ncu by 4.44% via using label correlation, and SMILE-Nc improves SMILE-Ncu by 33.33% via using unlabeled instances. In other words, by utilizing label correlation, SMILE on average can improve the performance of SMILE-Nc and SMILE-Ncu by 27.78%, and by exploiting unlabeled instances, SMILE on average can improve the performance of SMILE-Nu and SMILE-Ncu by 56.67%. From these results, we can draw a conclusion that both label correlation and unlabeled instances can be used to boost the performance of multi-label weak-label learning. These results justify our motivation to combine label correlation and unlabeled instances for multi-label classification using incomplete label information.

4.4. Runtime analysis

We also study the runtime cost of SMILE and the other comparing methods on the Cal500, Bibtex and Delicious datasets. All these methods are implemented with Matlab (R2013a 64 bit). The experiment platform is: Windows 10, Intel Core (TM) i5-4590 and 8GB RAM. The recorded runtime cost (average of 10 independent runs with 30% labeled instances with $m = 3$) of these comparing methods are listed in Table 11.

From Table 11, we have a clear overall observation that SMILE runs much faster than other comparing methods in almost all the cases. Taking running time results on Delicious for example, although ML-LOC and MLML assume the training instances are accurately labeled, they always take more time than the other methods (except ProDM). The reason is that ML-LOC employs a clustering algorithm to generate the LOC code, and MLML utilizes logistic regression with l_1 norm for each single label to initiate parameters. MLR-GL is another supervised multi-label approach, it relaxes the convex-concave optimization problem into a Second Order Cone Programming (SOCP) [18] problem, and it ranks 3rd (from fast to slow). Both ProDM and SMILE utilize abundant unlabeled instances in training process and need to solve matrix inverse problem, however, ProDM has much longer runtime than SMILE. This is because ProDM takes matrix inverse operation on an $N \times N$ matrix, while SMILE just needs to take matrix inverse operation on a $D \times D$ matrix, where $D \ll N$, especially for Bibtex and Delicious datasets. SSW solves the convex optimization problem via an approximated technique and ranks 4th. Tram and SMILE have a comparable runtime cost on Delicious dataset, this reason is that both Tram and SMILE construct a kNN graph and predict labels of unlabeled instances by solving a smoothness regularized problem.

Table 11
Statistic of runtime (in seconds).

Algorithm	Cal500	Bibtex	Delicious	Total
ML-LOC	7.8	530.78	9224.10	9762.68
MLR-GL	2.61	9.53	355.21	367.35
MLML	18.29	29.18	7567.86	7615.33
Tram	2.33	40.74	323.07	366.14
ProDM	3.34	110.55	8993.58	9107.47
SSW	1.76	48.76	411.38	461.90
SMILE	1.56	8.76	270.19	280.51

Given the superior effectiveness and efficiency of SMILE, it is desirable to use SMILE for multi-label classification with incomplete label information.

5. Conclusions

In this paper, we study multi-label classification using incomplete label information and propose an inductive approach called SMILE. SMILE jointly utilizes label correlation derived from labeled instances and abundant unlabeled instances to predict the labels of new unlabeled instances. We conducted experiments on three multi-label datasets, and compared our proposed approach with related methods. The experimental results show that SMILE performs significantly better than other related methods. In addition, we empirically studied the benefit of utilizing label correlation and abundant unlabeled instances and found combining them can boost the performance.

In our future work, we want to exploit high-order label correlation and design more efficient multi-label classification approaches for multi-label weak-label learning.

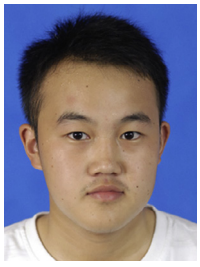
Acknowledgment

This work is supported by Natural Science Foundation of China (61402378), Natural Science Foundation of CQ CSTC (cstc2014jcyjA40031 and cstc2016jcyjA0351), Fundamental Research Funds for the Central Universities of China (2362015XK07 and XDJK2016B009), National Undergraduate Training Programs for Innovation and Entrepreneurship (201610635047), Southwest University Undergraduate Science and Technology Innovation Fund Project (20153601001).

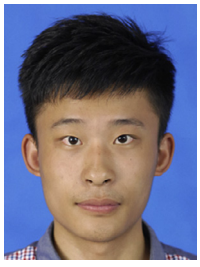
References

[1] G. Tsoumakas, I. Katakis, I. Vlahavas, Mining multi-label data, in: Data Mining and Knowledge Discovery Handbook, Springer, 2009, pp. 667–685.
[2] M.-L. Zhang, Z.-H. Zhou, A review on multi-label learning algorithms, IEEE Trans. Knowl. Data Eng. 26 (8) (2014) 1819–1837.
[3] A. Elisseeff, J. Weston, A kernel method for multi-labelled classification, in: Advances in Neural Information Processing Systems, 2001, pp. 681–687.
[4] A. McCallum, Multi-label text classification with a mixture model trained by em, in: Proceedings of the 16th AAAI Conference on Artificial Intelligence, Workshop on Text Learning, 1999, pp. 1–7.
[5] J. Read, B. Pfahringer, G. Holmes, E. Frank, Classifier chains for multi-label classification, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2009, pp. 254–269.
[6] M.-L. Zhang, K. Zhang, Multi-label learning by exploiting label dependency, in: Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2010, pp. 999–1008.
[7] G. Tsoumakas, I. Katakis, I. Vlahavas, Random k-labelsets for multilabel classification, IEEE Trans. Knowl. Data Eng. 23 (7) (2011) 1079–1089.
[8] S.-J. Huang, Z.-H. Zhou, Multi-label learning by exploiting label correlations locally, in: Proceedings of the 26th AAAI Conference on Artificial Intelligence, 2012, pp. 949–955.
[9] X. Zhu, Semi-Supervised Learning Literature Survey, Technical Report, Computer Sciences, University of Wisconsin-Madison, 2008.
[10] Z.-J. Zha, T. Mei, J. Wang, Z. Wang, X.-S. Hua, Graph-based semi-supervised learning with multiple labels, J. Vis Commun. Image Represent. 20 (2) (2009) 97–103.
[11] X. Kong, M.K. Ng, Z.-H. Zhou, Transductive multilabel learning via label set propagation, IEEE Trans. Knowl. Data Eng. 25 (3) (2013) 704–719.
[12] Y. Guo, D. Schuurmans, Semi-supervised multi-label classification, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2012, pp. 355–370.
[13] Y. Guo, D. Schuurmans, Adaptive large margin training for multilabel classification, in: Proceedings of 25th AAAI Conference on Artificial Intelligence, 2011.
[14] X. Zhang, Y. Yu, M. White, R. Huang, D. Schuurmans, Convex sparse coding, subspace learning, and semi-supervised extensions, in: Proceedings of the 25th AAAI Conference on Artificial Intelligence, 2011, pp. 567–573.
[15] L. Wu, M.-L. Zhang, Multi-label classification with unlabeled data: An inductive approach., in: Preceedings of the 5th Asian Conference on Machine Learning, 2013, pp. 197–212.
[16] L. Jing, L. Yang, J. Yu, M.K. Ng, Semi-supervised low-rank mapping learning for multi-label classification, in: Proceedings of the 25th IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1483–1491.

[17] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Found. Trends Mach. Learn. 3 (1) (2011) 1–122.
[18] S.S. Bucak, R. Jin, A.K. Jain, Multi-label learning with incomplete class assignments, in: Proceedings of 24th IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 2801–2808.
[19] Z. Qi, M. Yang, Z.M. Zhang, Z. Zhang, Mining partially annotated images, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2011, pp. 1199–1207.
[20] X. Kong, Z. Wu, L.-J. Li, R. Zhang, H. Wu, W. Fan, Large-scale multi-label learning with incomplete label assignments, in: SIAM International Conference on Data Mining, 2014, pp. 920–928.
[21] Y.-Y. Sun, Y. Zhang, Z.-H. Zhou, Multi-label learning with weak label, in: Proceedings of the 24th AAAI Conference on Artificial Intelligence, 2010.
[22] G. Yu, G. Zhang, H. Rangwala, C. Domeniconi, Z. Yu, Protein function prediction using weak-label learning, in: Proceedings of the ACM Conference on Bioinformatics, Computational Biology and Biomedicine, 2012, pp. 202–209.
[23] G. Yu, C. Domeniconi, H. Rangwala, G. Zhang, Protein function prediction using dependence maximization, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases, Springer, 2013, pp. 574–589.
[24] B. Wu, Z. Liu, S. Wang, B.-G. Hu, Q. Ji, Multi-label learning with missing labels, in: Proceedings of the 22nd International Conference on Pattern Recognition, 2014, pp. 1964–1968.
[25] B. Wu, S. Lyu, B. Ghanem, ML-mg: Multi-label learning with missing labels using a mixed graph, in: Proceedings of the 25th IEEE International Conference on Computer Vision, 2015, pp. 4157–4165.
[26] H.-F. Yu, P. Jain, P. Kar, I.S. Dhillon, Large-scale multi-label learning with missing labels, in: Proceedings of the 31st International Conference on Machine Learning, 2014, pp. 593–601.
[27] B. Wu, S. Lyu, B.-G. Hu, Q. Ji, Multi-label learning with missing labels for image annotation and facial action unit recognition, Pattern Recognit. 48 (7) (2015) 2279–2289.
[28] X. Li, F. Zhao, Y. Guo, Conditional restricted Boltzmann machines for multi-label learning with incomplete labels, in: Proceedings of the 16th International Conference on Artificial Intelligence and Statistics, 2015, pp. 635–643.
[29] P. Smolensky, Information processing in dynamical systems: foundations of harmony theory, in: Parallel distributed processing: Explorations in the Microstructure of Cognition, 1986, pp. 194–281.
[30] A. Panchenko, C. Fairon, A study of heterogeneous similarity measures for semantic relation extraction, in: Preceedings of the Joint Conference JEP-TALN-RECITAL, 2012.
[31] Y. Liu, R. Jin, L. Yang, Semi-supervised multi-label learning by constrained non-negative matrix factorization, in: Proceedings of the 21th National Conference on Artificial Intelligence, 2006, pp. 421–426.
[32] Z. Lu, Y. Peng, Exhaustive and efficient constraint propagation: a graph-based learning approach and its applications, Int. J. Comput. Vis. (2013) 306–325.
[33] D.Y. Hu, L. Reichel, Krylov-subspace methods for the Sylvester equation, Linear Algebra Appl. 172 (7) (1992) 283–313.
[34] A. Gretton, O. Bousquet, A. Smola, B. Schölkopf, Measuring statistical dependence with Hilbert-Schmidt norms, in: International Conference Algorithmic Learning Theory, Springer, 2005, pp. 63–77.
[35] F. Zhao, Y. Guo, Semi-supervised multi-label learning with incomplete labels, in: Proceedings of the 24th International Conference on Artificial Intelligence, 2015, pp. 4062–4068.
[36] M. Belkin, P. Niyogi, V. Sindhwani, Manifold regularization: a geometric framework for learning from labeled and unlabeled examples, J. Mach. Learn. Res. 7 (2006) 2399–2434.
[37] N. Ghamrawi, A. McCallum, Collective multi-label classification, in: Proceedings of the 14th ACM International Conference on Information and Knowledge Management, 2005, pp. 195–200.
[38] G.-J. Qi, X.-S. Hua, Y. Rui, J. Tang, T. Mei, H.-J. Zhang, Correlative multi-label video annotation, in: Proceedings of the 15th ACM International Conference on Multimedia, 2007, pp. 17–26.
[39] F. Kang, R. Jin, R. Sukthankar, Correlated label propagation with application to multi-label learning, in: Proceedings of the 19th IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 1719–1726.
[40] Z.-J. Zha, T. Mei, Z. Wang, X.-S. Hua, Building a comprehensive ontology to refine video concept detection, in: Proceedings of the 9th International Workshop on Multimedia Information Retrieval, 2007, pp. 227–236.
[41] G. Yu, G. Zhang, Z. Zhang, Z. Yu, Semi-supervised classification based on subspace sparse representation, Knowl. Information Syst. 43 (1) (2015) 81–101.
[42] G. Chen, Y. Song, F. Wang, C. Zhang, Semi-supervised multi-label learning by solving a Sylvester equation, in: Proceedings of the 8th SIAM International Conference on Data Mining, 2008, pp. 410–419.
[43] F.R. Chung, Spectral Graph Theory, 92, American Mathematical Soc, 1997.
[44] D. Zhou, O. Bousquet, T.N. Lal, J. Weston, B. Schölkopf, Learning with local and global consistency, in: Advances in Neural Information Processing Systems, 2003, pp. 321–328.
[45] S. Arya, D.M. Mount, N.S. Netanyahu, R. Silverman, A.Y. Wu, An optimal algorithm for approximate nearest neighbor searching fixed dimensions, J. ACM 45 (6) (1998) 891–923.
[46] J. Nocedal, S. Wright, Numerical Optimization, Springer Verlag, 1999.
[47] D. Turnbull, L. Barrington, D. Torres, G. Lanckriet, Semantic annotation and retrieval of music and sound effects, IEEE Trans. Audio Speech Lang. Process. 16 (2) (2008) 467–476.
[48] I. Katakis, G. Tsoumakas, I. Vlahavas, Multilabel text classification for automated tag suggestion, in: Joint European Conference on Machine Learning and Knowledge Discovery in Database in Discovery Challenge, 2008, pp. 75–83.
[49] G. Tsoumakas, I. Katakis, I. Vlahavas, Effective and efficient multilabel classification in domains with large number of labels, in: Joint European Conference on Machine Learning and Knowledge Discovery in Databases in Mining Multidimensional Data, 2008, pp. 30–44.
[50] Y. Zhang, Z.-H. Zhou, Multilabel dimensionality reduction via dependence maximization, ACM Trans. Knowl. Discov. Data 4 (3) (2010) 1–21.



Qiaoyu Tan is a undergraduate student at the College of Computer and Information Science, Southwest University, Chongqing, China. His research is supported by National Undergraduate Training Programs for Innovation and Entrepreneurship. His current research interests include machine learning and data mining, especially semi-supervised learning and multi-label learning.



Yanmin Yu is a undergraduate student at the College of Computer and Information Science, Southwest University, Chongqing, China. His research is supported by National Undergraduate Training Programs for Innovation and Entrepreneurship. His research interests include semi-supervised learning and dimensionality reduction.

Guoxian Yu is an Associate Professor in the College of Computer and Information Science, Southwest University, Chongqing, China. He received the Ph.D. in Computer Science from South China University of Technology, Guangzhou, China in 2013. He was a Postdoc Research Fellow at the Department of Computer Science, Hong Kong Baptist University, Hong Kong from 2014 to 2015. His current research interests include machine learning and bioinformatics. He has served as PC member for KDD, ICDM, SDM and other conferences. He is a recipient of Best Poster Award of SDM2012 Doctral Forum and Best Student Paper Award of IEEE International Conference on Machine Learning and Cybernetics, 2011.

Jun Wang is an Associate Professor in the College of Computer and Information Science, Southwest University, Chongqing, China. She received B.Sc. degree in Computer Science, M.Eng. degree in Computer Science and Ph.D. in Artificial Intelligence from Harbin Institute of Technology, Harbin, China in 2004, 2006 and 2010, respectively. Her current research interests include machine learning, data mining and their applications in bioinformatics.