Journal of Systems Engineering and Electronics
Vol. 33, No. 5, October 2022, pp.1043 – 1051

Label correlation for partial label learning

GE Lingchi, FANG Min*, LI Haikun, and CHEN Bo

School of Computer Science and Technology, Xidian University, Xi'an 710071, China

Abstract: Partial label learning aims to learn a multi-class classifier, where each training example corresponds to a set of candidate labels among which only one is correct. Most studies in the label space have only focused on the difference between candidate labels and non-candidate labels. So far, however, there has been little discussion about the label correlation in the partial label learning. This paper begins with a research on the label correlation, followed by the establishment of a unified framework that integrates the label correlation, the adaptive graph, and the semantic difference maximization criterion. This work generates fresh insight into the acquisition of the learning information from the label space. Specifically, the label correlation is calculated from the candidate label set and is utilized to obtain the similarity of each pair of instances in the label space. After that, the labeling confidence for each instance is updated by the smoothness assumption that two instances should be similar outputs in the label space if they are close in the feature space. At last, an effective optimization program is utilized to solve the unified framework. Extensive experiments on artificial and real-world data sets indicate the superiority of our proposed method to state-of-art partial label learning methods.

Keywords: pattern recognition, partial label learning, label correlation, disambiguation.

DOI: 10.23919/JSEE.2022.000102

1. Introduction

Although learning from the training examples associated with accurate labels is effective, collecting such labeled data is expensive in many real-world classification tasks. The aim of partial label learning is to learn a multi-class classifier from ambiguously labeled examples which can be easily obtained. Recently, partial label learning has arisen in many real-world applications, such as ecoinformatics [1], natural language processing [2], and automatic image annotation [3,4].

The difficulty of partial label learning is that the ground-truth label hidden in the candidate label set cannot be accessed by the learning algorithms directly. The candidate label disambiguation is a straightforward method to deal with partial label learning. The current disambiguation-based approaches can be divided into two categories: the averaging-based strategy and the identification-based strategy. For the averaging-based strategy, the candidate labels of each training example are treated equally and the predictive model is made by averaging their modeling output [5,6]. For the identification-based strategy, the ground-truth label is regarded as a latent variable which can be determined by an iterative refining procedure [7−12].

It is well-known that the label correlation has a pivotal role in multi-label learning [13]. So far, little attention has been paid to the role of label correlation in the multi-class classification because of a lack of the label correlation information in the label space. However, partial label learning could be a contributing factor to build a multi-class classifier by the label correlation. Specifically, if a pair of classes is hard to distinguish, they are easy to appear in the candidate label set of the same sample in the partial label learning. In terms of this agreement, label correlation can be built from the label space. After that, label correlation is utilized to disambiguate candidate label sets by the smoothness assumption that two instances should be the similar output in the label space if they are close in the feature space.

In the label space, the global label information can be extracted by label correlation. The label relationship at the instance level can be extracted by the semantic difference maximization criterion [12]. At last, to overcome the influence of noise and outliers in the feature space, the adaptive graph [10] can be integrated into the unified framework. Then, a novel approach named label correlation, semantic difference maximization, and adaptive graph for partial label learning (PL-LCSA) is proposed in this paper.

There are two innovations in this paper: (i) To the best of our knowledge, this paper is the first one that label cor-

Manuscript received May 10, 2021.
*Corresponding author.
This work was supported by the National Natural Science Foundation of China (62176197;61806155) and the National Natural Science Foundation of Shaanxi Province (2020GY-062).

relation is introduced to deal with partial label learning, and the information from the label space can be further expanded. (ii) Label correlation, semantic difference maximization, and the adaptive graph are integrated into a learning framework, in which the information from the label space and the feature space can be learned simultaneously. Comprehensive experiments show that PL-LCSA achieves competitive performance against the state-of-the-art partial label learning approaches in the artificial and real-world data sets.

2. Related work

Formally, let $\mathcal{X}=\mathbf{R}^d$ be the d-dimensional feature space and $\mathcal{Y}=\{0,1\}^l$ be the label space with l class labels. The training set can be denoted by $\mathcal{D}=\{(\boldsymbol{x}_i, Z_i)|1\leqslant i\leqslant m\}$, where $\boldsymbol{x}_i\in\mathcal{X}$ is a d-dimensional feature vector and $Z_i\in\mathcal{Y}$ is the candidate label set associated with $\boldsymbol{x}_i$. The ground-truth label $\boldsymbol{y}_i^*$ associated with $\boldsymbol{x}_i$ is assumed to reside in the candidate label set Z_i, i.e., $\boldsymbol{y}_i^*\in Z_i$. The aim of the partial label learning is to learn a multi-class classifier f: $\mathcal{X}\rightarrow\mathcal{Y}$ from the training set $\mathcal{D}$.

The ground-truth label which resides in the candidate label set is not directly accessible to the learning algorithm. Disambiguation is a major approach to recover the ground-truth labeling information. There are currently two main disambiguation strategies, i.e., the averaging-based strategy and the identification-based strategy.

The averaging-based strategy is to treat the candidate labels in an identical manner and predict unseen instances by averaging the output of candidate labels. The k-nearest neighbor technique for partial label learning was proposed in [5], and the candidate label set of neighbor instances are integrated by weighted voting to make the prediction of unseen instances, i.e., $\arg\max_{y\in\mathcal{Y}}\sum_{\boldsymbol{x}_i\in N_k(\boldsymbol{x}_i)}\boldsymbol{w}_j\mathrm{II}(y\in Z_j)$ ($N_k(\boldsymbol{x}_i)$ is the set of k-nearest neighbor for $\boldsymbol{x}_i$). The deficiency of the averaging-based strategy is that the output of the ground-truth label will be overwhelmed by false positive labels.

To overcome the weakness of the averaging-based strategy, the identification-based strategy is to make the ground-truth label a latent variable which can be identified by an iterative process. There are several techniques to iteratively refine the ground-truth labeling confidences, e.g., the maximum latent semantic differences criterion [14], the maximum-likelihood technique [15], the maximum margin criterion [7,16], the dictionary-based learning criterion [17], the boosting technique [18], the heterogeneous loss with sparse and low-rank regularization [19], artificial neural network technology [4,20].

Recently, the feature-aware disambiguation strategy aims to disambiguate the candidate label set by the information from the feature space. The reconstructing information from the k-nearest neighbor is used to update the labeling confidence matrix iteratively [8]. The manifold structure of the feature space is propagated to the label space for disambiguation [9]. To overcome the noise and outliers in the feature space, the adaptive graph is utilized to guide disambiguation [11]. In the label space, the semantic difference maximization criterion aims to maximize the latent semantic differences of two instances which do not share any common candidate labels [14].

Different from the method of disambiguation, some algorithms work by binary decomposition for partial label learning. The disambiguation-free approach learns the predictive model by the error-correcting output codes (ECOC) technique [21]. The One-vs-One decomposition strategy is adapted to solving the partial label learning problem [22].

Most studies in the field of the label space have only focused on the difference between candidate labels and non-candidate labels, and ignored the label correlation. In the next section, we present a novel approach which introduces the label correlation to partial label learning and simultaneously utilizes the information from the feature space and the label space to disambiguate the candidate label set.

3. Approach

In this section, we introduce the proposed approach PL-LCSA. Firstly, we present the label correlation and the uniform learning framework. After that, an effective optimization procedure is utilized to deal with this framework.

3.1 Label correlation

We denote $\boldsymbol{X}=[\boldsymbol{x}_1, \boldsymbol{x}_2,\cdots, \boldsymbol{x}_m]^{\mathrm{T}}\in\mathbf{R}^{m\times d}$ as the feature matrix, and $\boldsymbol{Y}=[\boldsymbol{y}_1, \boldsymbol{y}_2,\cdots, \boldsymbol{y}_m]^{\mathrm{T}}\in\mathbf{R}^{m\times l}$ as the partial label matrix where $y_{i,j}=1$ means that the j-label is one of the candidate labels of $\boldsymbol{x}_i$, i.e., $y_{i,j}\in Z_i$. Otherwise, $y_{i,j}=0$. Let $\boldsymbol{P}=[p_{ij}]_{m\times l}$ be the labeling confidence matrix, where m denotes the number of training examples, l denotes the number of class labels, and $p_{i,j}$ indicates the probability of the jth label as the ground-truth label associated with instance $\boldsymbol{x}_i$. For each training example $(\boldsymbol{x}_i, Z_i)$, we aim to generate the labeling confidence vector $\boldsymbol{p}_i=[p_{i1}, p_{i2},\cdots, p_{il}]$. $\boldsymbol{P}$ should be constrained to the following condition:

$$\sum_{j=1}^{l}p_{i,j}=1,\ \ \forall i\in\{1,2,\cdots,m\},\tag{1}$$

$$0\leqslant\boldsymbol{p}_i\leqslant\boldsymbol{y}_i,\ \ \forall i\in\{1,2,\cdots,m\}.\tag{2}$$

The aim of (1) is to normalize $\boldsymbol{p}_i$, and make the sum of

labeling confidence vector $\boldsymbol{p}_i$ be 1 for each instance. The aim of (2) is to make sure that the labeling confidence of non-candidate labels must be 0, and the ground-truth label resides in the candidate label set. Once the labeling confidence vectors are normalized, the partial label training set $\mathcal{D}$ can be transformed into its disambiguation counterpart $\mathcal{D}=\{(\boldsymbol{x}_i, \boldsymbol{p}_i)|1\leqslant i\leqslant m\}$. Then the predictive model can be learned by the disambiguation results $\mathcal{D}$.

The label correlation matrix can be denoted as $\boldsymbol{B}\in\mathbf{R}^{l\times l}$, where $b_{i,j}$ is the similarity between the i th and the jth labels. The label correlation matrix can be calculated by the cosine similarity:

$$b_{i,j}=\cos(\boldsymbol{q}_i,\boldsymbol{q}_j)=\frac{\langle \boldsymbol{q}_i,\boldsymbol{q}_j\rangle}{\|\boldsymbol{q}_i\|\|\boldsymbol{q}_j\|}=\frac{\sum_{s=1}^{m}\boldsymbol{q}_{i,s}\boldsymbol{q}_{j,s}}{\sqrt{\sum_{s=1}^{m}\boldsymbol{q}_{i,s}^2}\sqrt{\sum_{s=1}^{m}\boldsymbol{q}_{j,s}^2}} \tag{3}$$

where $\boldsymbol{q}_i$ is the i th column of label matrix $\boldsymbol{Y}=[\boldsymbol{y}_1, \boldsymbol{y}_2, \cdots, \boldsymbol{y}_m]^{\mathrm{T}}$, i.e., $\boldsymbol{Y}=[\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_l]$.

To disambiguate the candidate label set by the label correlation, the labeling confidence matrix will be generated by the smoothness assumption that two points should be the similar output in the label space if they are close in the feature space.

According to this assumption, we can calculate the similarity of instances in the label space by

$$\mathrm{sim}L_{i,j}=\boldsymbol{p}_i^{\mathrm{T}}\boldsymbol{B}\boldsymbol{p}_j=\sum_{k_1=1}^{l}\sum_{k_2=1}^{l}p_{i,k_1}b_{k_1,k_2}p_{j,k_2} \tag{4}$$

where $\mathrm{sim}L_{i,j}$ is the similarity between instance $\boldsymbol{x}_i$ and instance $\boldsymbol{x}_j$ in the label space. The larger the similarity is, the bigger $\mathrm{sim}L_{i,j}$ is.

Then the similarity of instances in the feature space can be obtained according to

$$h_{i,j}=\begin{cases}\exp\left(-\dfrac{\|\boldsymbol{x}_i-\boldsymbol{x}_j\|^2}{2\sigma^2}\right), & \boldsymbol{x}_i\in N_k(\boldsymbol{x}_j)\\ 0, & \boldsymbol{x}_i\notin N_k(\boldsymbol{x}_j)\end{cases} \tag{5}$$

where σ is the average Euclidean distance among each pair of instances, i.e., $\sigma=\left(\sum_{i=1}^{m}\sum_{j=1}^{m}\|\boldsymbol{x}_i-\boldsymbol{x}_j\|_2\right)\cdot m^{-2}$. $h_{i,j}$ is the similarity between instance $\boldsymbol{x}_i$ and instance $\boldsymbol{x}_j$ in the feature space.

Then we should model the smoothness assumption by minimizing the gap of instances similarity between the label space and the feature space, and the label confidence matrix can be calculated by

$$\begin{aligned}&\min_{\boldsymbol{P}}\sum_{i=1}^{m}\sum_{j=1}^{m}\left(h_{i,j}-\boldsymbol{p}_i^{\mathrm{T}}\boldsymbol{B}\boldsymbol{p}_j\right)^2,\\ &\text{s.t. }\boldsymbol{P}\mathbf{1}_l=\mathbf{1}_m,\ \mathbf{0}_{m\times l}\leqslant\boldsymbol{P}\leqslant\boldsymbol{Y},\end{aligned} \tag{6}$$

where $\mathbf{1}_l=[1,1,\cdots,1]^{\mathrm{T}}\in\mathbf{R}^l$ is an all 1 vector with size l, $\mathbf{0}_{m\times l}$ is an all 0 matrix with size $m\times l$, and constraints are the matrix form of (1) and (2) respectively.

3.2 Uniform learning framework

The feature and label information are simultaneously utilized to disambiguate the candidate label sets in our uniform learning framework.

In the feature space, we adopt the adaptive graph to recover the intrinsic manifold structure within the data more robustly and accurately. Let $\mathcal{G}=(\mathcal{N},\Xi,\mathcal{S})$ be a weighted graph, where $\mathcal{N}=\{\boldsymbol{x}_i|1\leqslant i\leqslant m\}$ is the node of the graph, and $\Xi=\{\{\boldsymbol{x}_i,\boldsymbol{x}_j\}|\boldsymbol{x}_j\in N_k(\boldsymbol{x}_i),1\leqslant i\leqslant m\}$ is a set of edges from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ if and only if $\boldsymbol{x}_j$ is among the k-nearest neighbors of $\boldsymbol{x}_i$. $\mathcal{S}\in\mathbf{R}^{m\times m}$ corresponds to the information of the manifold structure in the feature space. $\boldsymbol{L}\in\mathbf{R}^{m\times m}$ is an index matrix where $l_{i,j}=1$ if $(\boldsymbol{x}_i,\boldsymbol{x}_j)\in\Xi$; otherwise, $l_{i,j}=0$. The labeling confidence matrix can be calculated [11] by solving the following problem:

$$\begin{aligned}\min_{\boldsymbol{P},\boldsymbol{S},\boldsymbol{W}}&\sum_{i=1}^{m}\|f(\boldsymbol{x}_i,\boldsymbol{W})-\boldsymbol{p}_i\|_2^2+\alpha\sum_{i=1}^{m}\|\boldsymbol{p}_i-\sum_{l_{i,j}=1}s_{i,j}\boldsymbol{p}_j\|_2^2+\\ &\beta\sum_{i=1}^{m}\|\boldsymbol{x}_i-\sum_{l_{i,j}=1}s_{i,j}\boldsymbol{x}_j\|_2^2+\lambda\|\boldsymbol{W}\|_{\mathrm{F}}^2,\\ &\text{s.t.}\begin{cases}\boldsymbol{P}\mathbf{1}_l=\mathbf{1}_m\\ \mathbf{0}_{m\times l}\leqslant\boldsymbol{P}\leqslant\boldsymbol{Y}\\ \boldsymbol{S}\mathbf{1}_m=\mathbf{1}_m\\ \mathbf{0}_{m\times m}\leqslant\boldsymbol{S}\leqslant\boldsymbol{L}\end{cases},\end{aligned} \tag{7}$$

where $f(\boldsymbol{x}_i,\boldsymbol{W})$ is a predictive model, and $\boldsymbol{W}$ is the model parameter. The first term is to learn $\boldsymbol{W}$. The second term is to learn the labeling confidence matrix, the third term is to determine the adaptive graph weight matrix, and the fourth term is a regularization term.

In the label space, PL-LCSA aims to disambiguate candidate label sets by the label correlation and the semantic difference maximization. The aim of semantic difference maximization is to maximize the semantic differences of the two instances which do not share any common candidate labels. It can be expressed [14] by the following problem:

$$\max_{\boldsymbol{P}}\sum_{i=1}^{m}\sum_{j=1}^{m}r_{ij}\left\|\boldsymbol{p}_i-\boldsymbol{p}_j\right\|_2^2 \tag{8}$$

where $\boldsymbol{R}=[r_{i,j}]_{m\times m}$ is an index matrix; $r_{i,j}=1$ if $\boldsymbol{Y}_i^{\mathrm{T}}\boldsymbol{Y}_j=0$, otherwise $r_{i,j}=0$.

By integrating (6), (7), and (8), the final objective

function is shown as follows:

$$\min_{\boldsymbol{P},\boldsymbol{S},\boldsymbol{W}} \sum_{i=1}^{m} \|f(\boldsymbol{x}_i,\boldsymbol{W})-\boldsymbol{p}_i\|_2^2+\delta\sum_{i=1}^{m}\sum_{j=1}^{m}\left(h_{i,j}-\boldsymbol{p}_i^{\mathrm{T}}\boldsymbol{B}\boldsymbol{p}_j\right)^2+$$
$$\alpha\sum_{i=1}^{m}\|\boldsymbol{p}_i-\sum_{l_{i,j}=1}s_{i,j}\boldsymbol{p}_j\|_2^2+\beta\sum_{i=1}^{m}\|\boldsymbol{x}_i-\sum_{l_{i,j}=1}s_{i,j}\boldsymbol{x}_j\|_2^2-$$
$$\gamma\sum_{i=1}^{m}\sum_{j=1}^{m}r_{i,j}\|\boldsymbol{p}_i-\boldsymbol{p}_j\|_2^2+\lambda\|\boldsymbol{W}\|_{\mathrm{F}}^2,$$
$$\text{s.t.}\begin{cases}\boldsymbol{P}\boldsymbol{1}_l=\boldsymbol{1}_m\\ \boldsymbol{0}_{m\times l}\leqslant\boldsymbol{P}\leqslant\boldsymbol{Y}\\ \boldsymbol{S}\boldsymbol{1}_m=\boldsymbol{1}_m\\ \boldsymbol{0}_{m\times m}\leqslant\boldsymbol{S}\leqslant\boldsymbol{L}\end{cases}, \tag{9}$$

where λ is the regularization parameter, δ, α, β, and γ are the trade-off parameters.

3.3 Alternative optimization

Before the optimization procedure, the matrices $\boldsymbol{P}$ and $\boldsymbol{S}$ should be initialized by solving the standard quadratic programming (QP) problems [11] as follows:

$$\min_{\boldsymbol{S}}\sum_{i=1}^{m}\|\boldsymbol{x}_i-\sum_{l_{i,j}=1}s_{i,j}\boldsymbol{x}_j\|_2^2,$$
$$\text{s.t.}\begin{cases}\boldsymbol{S}\boldsymbol{1}_m=\boldsymbol{1}_m\\ \boldsymbol{0}_{m\times m}\leqslant\boldsymbol{S}\leqslant\boldsymbol{L}\end{cases}, \tag{10}$$

$$\min_{\boldsymbol{P}}\sum_{i=1}^{m}\|\boldsymbol{p}_i-\sum_{l_{i,j}=1}s_{i,j}\boldsymbol{p}_j\|_2^2,$$
$$\text{s.t.}\begin{cases}\boldsymbol{P}\boldsymbol{1}_l=\boldsymbol{1}_m\\ \boldsymbol{0}_{m\times l}\leqslant\boldsymbol{P}\leqslant\boldsymbol{Y}\end{cases}. \tag{11}$$

Then, we utilize the alternative optimization procedure to solve the problem (9).

Update $\boldsymbol{S}$, with fixed $\boldsymbol{P}$ and $\boldsymbol{W}$, the problem (9) can be stated as follows:

$$\min_{\boldsymbol{S}}\alpha\sum_{i=1}^{m}\left\|\boldsymbol{p}_i-\sum_{l_{i,j}=1}s_{i,j}\boldsymbol{p}_j\right\|_2^2+$$
$$\beta\sum_{i=1}^{m}\left\|\boldsymbol{x}_i-\sum_{l_{i,j}=1}s_{i,j}\boldsymbol{x}_j\right\|_2^2$$
$$\text{s.t.}\begin{cases}\boldsymbol{S}\boldsymbol{1}_m=\boldsymbol{1}_m\\ \boldsymbol{0}_{m\times m}\leqslant\boldsymbol{S}\leqslant\boldsymbol{L}\end{cases}. \tag{12}$$

The optimization problem (12) can be rewritten [11] as follows:

$$\min_{\hat{\boldsymbol{s}}_i}\tilde{\boldsymbol{s}}_i^{\mathrm{T}}(\alpha(\boldsymbol{D}_i^p)^{\mathrm{T}}(\boldsymbol{D}_i^p)+\beta(\boldsymbol{D}_i^x)^{\mathrm{T}}(\boldsymbol{D}_i^x))\tilde{\boldsymbol{s}}_i,$$
$$\text{s.t.}\begin{cases}\boldsymbol{1}_k^{\mathrm{T}}\tilde{\boldsymbol{s}}_i=1\\ \boldsymbol{0}_k\leqslant\tilde{\boldsymbol{s}}_i\leqslant\boldsymbol{1}_k\end{cases}, \tag{13}$$

where $\tilde{\boldsymbol{s}}_i\in\mathbf{R}^k$ denotes the weight vector which shows the importance of the neighbor sample in reconstructing $\boldsymbol{x}_i$. Denote matrix $\boldsymbol{D}_i^p=[\boldsymbol{p}_i-\boldsymbol{p}_i^1,\boldsymbol{p}_i-\boldsymbol{p}_i^2,\cdots,\boldsymbol{p}_i-\boldsymbol{p}_i^k]^{\mathrm{T}}\in\mathbf{R}^{k\times l}$ and $\boldsymbol{D}_i^x=[\boldsymbol{x}_i-\boldsymbol{x}_i^1,\boldsymbol{x}_i-\boldsymbol{x}_i^2,\cdots,\boldsymbol{x}_i-\boldsymbol{x}_i^k]^{\mathrm{T}}\in\mathbf{R}^{k\times d}$, where $\boldsymbol{p}_i^j$ is the labeling confidence vector associated with the jth nearest neighbors of $\boldsymbol{x}_i$, and $\boldsymbol{x}_i^j$ is the jth nearest neighbors of $\boldsymbol{x}_i$. The optimization problem (13) is a standard QP problem which can be solved by existing QP tools. We adopt interior point methods to solve (13) by the quadprag function in Matlab.

Updating $\boldsymbol{P}$, with fixed $\boldsymbol{S}$ and $\boldsymbol{W}$, the problem (9) can be stated as follows:

$$\min_{\boldsymbol{P}}\sum_{i=1}^{m}\|f(\boldsymbol{x}_i,\boldsymbol{W})-\boldsymbol{p}_i\|_2^2+\delta\sum_{i=1}^{m}\sum_{j=1}^{m}\left(h_{i,j}-\boldsymbol{p}_i^{\mathrm{T}}\boldsymbol{B}\boldsymbol{p}_j\right)^2+$$
$$\alpha\sum_{i=1}^{m}\left\|\boldsymbol{p}_i-\sum_{l_{i,j}=1}s_{i,j}\boldsymbol{p}_j\right\|_2^2-\gamma\sum_{i=1}^{m}\sum_{j=1}^{m}r_{i,j}\|\boldsymbol{p}_i-\boldsymbol{p}_j\|_2^2,$$
$$\text{s.t.}\begin{cases}\boldsymbol{P}\boldsymbol{1}_l=\boldsymbol{1}_m\\ \boldsymbol{0}_{m\times l}\leqslant\boldsymbol{P}\leqslant\boldsymbol{Y}\end{cases}. \tag{14}$$

In the optimization problem (14), the first three terms are convex, and the last term is concave. To optimize the second term, the first order Taylor approximation at $\boldsymbol{P}_0$ can be utilized to replace it.

$$\sum_{i=1}^{m}\sum_{j=1}^{m}\left(h_{i,j}-\boldsymbol{p}_i^{\mathrm{T}}\boldsymbol{B}\boldsymbol{p}_j\right)^2=\|\boldsymbol{H}-\boldsymbol{P}\boldsymbol{B}\boldsymbol{P}^{\mathrm{T}}\|_{\mathrm{F}}^2\approx$$
$$\mathrm{tr}((\boldsymbol{H}-\boldsymbol{P}_0{}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{P}_0)^{\mathrm{T}}(\boldsymbol{H}-\boldsymbol{P}_0{}^{\mathrm{T}}\boldsymbol{B}\boldsymbol{P}_0))+$$
$$\mathrm{tr}((-2\boldsymbol{H}\boldsymbol{P}_0\boldsymbol{B}^{\mathrm{T}}-2\boldsymbol{H}^{\mathrm{T}}\boldsymbol{P}_0\boldsymbol{B}+2\boldsymbol{P}_0\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}_0{}^{\mathrm{T}}\boldsymbol{P}_0\boldsymbol{B}+$$
$$2\boldsymbol{P}_0\boldsymbol{B}\boldsymbol{P}_0{}^{\mathrm{T}}\boldsymbol{P}_0\boldsymbol{B}^{\mathrm{T}})^{\mathrm{T}}(\boldsymbol{P}-\boldsymbol{P}_0))$$

where $\boldsymbol{P}_0$ is the updated value at the previous iteration of $\boldsymbol{P}$. After Taylor approximation of the second term, the first three terms are still convex, and the last term is concave. The problem (14) is a constrained convex-concave problem. Fortunately, the concave-convex procedure (CCCP) can be used to solve this optimization problem [23]. A rigorous analysis of the convergence of CCCP is provided by [24]. The idea of CCCP is to linearize the concave part of the objective function. Therefore, the last term can be linearized by its first order Taylor approximation:

$$-\sum_{i=1}^{m}\sum_{j=1}^{m}r_{i,j}\|\boldsymbol{p}_i-\boldsymbol{p}_j\|_2^2=-\mathrm{tr}(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{L}_r\boldsymbol{P})\approx$$
$$-\mathrm{tr}(\boldsymbol{P}_0^{\mathrm{T}}\boldsymbol{L}_r\boldsymbol{P}_0+(\boldsymbol{L}_r\boldsymbol{P}_0+\boldsymbol{L}_r{}^{\mathrm{T}}\boldsymbol{P}_0)^{\mathrm{T}}(\boldsymbol{P}-\boldsymbol{P}_0))$$

where $\boldsymbol{L}_r=\mathrm{diag}(\boldsymbol{R}_a)-\boldsymbol{R}$ is a Laplacian matrix where vector $\boldsymbol{R}_a$ is the sum of row for matrix $\boldsymbol{R}$.

We define the modeling output matrix $\boldsymbol{F}=[f(\boldsymbol{x}_1,\boldsymbol{W}),f(\boldsymbol{x}_2,\boldsymbol{W}),\cdots,f(\boldsymbol{x}_m,\boldsymbol{W})]^{\mathrm{T}}$. In order to simplify notation, we define $\boldsymbol{V}=\boldsymbol{L}_r\boldsymbol{P}_0+\boldsymbol{L}_r{}^{\mathrm{T}}\boldsymbol{P}_0$ and $\boldsymbol{U}=-2\boldsymbol{H}\boldsymbol{P}_0\boldsymbol{B}^{\mathrm{T}}-2\boldsymbol{H}^{\mathrm{T}}\boldsymbol{P}_0\boldsymbol{B}+2\boldsymbol{P}_0\boldsymbol{B}^{\mathrm{T}}\boldsymbol{P}_0{}^{\mathrm{T}}\boldsymbol{P}_0\boldsymbol{B}+2\boldsymbol{P}_0\boldsymbol{B}\boldsymbol{P}_0{}^{\mathrm{T}}\boldsymbol{P}_0\boldsymbol{B}^{\mathrm{T}}$. Then the optimization prob-

lem (14) can be rewritten as follows:

$$\min_{\widetilde{\boldsymbol{p}}} \frac{1}{2}\widetilde{\boldsymbol{p}}^{\mathrm{T}}(\boldsymbol{\Theta}+\boldsymbol{\Theta}^{\mathrm{T}})\widetilde{\boldsymbol{p}}+(\delta\widetilde{\boldsymbol{u}}-2\widetilde{\boldsymbol{f}}-\widetilde{\boldsymbol{v}})^{\mathrm{T}}\widetilde{\boldsymbol{p}},$$
$$\text{s.t.}\begin{cases}\boldsymbol{0}\leqslant\widetilde{\boldsymbol{p}}\leqslant\widetilde{\boldsymbol{y}}\\ \sum\limits_{i=1,i\%m=j}^{ml}\widetilde{\boldsymbol{p}}_i=1\\ 0\leqslant j\leqslant m-1\end{cases}, \tag{15}$$

where $\tilde{\boldsymbol{p}}$ is the vectorization of matrix $\boldsymbol{P}$, i.e., $\tilde{\boldsymbol{p}}=\mathrm{vec}(\boldsymbol{P})\in\mathbf{R}^{ml}$, $\tilde{\boldsymbol{f}}=\mathrm{vec}(\boldsymbol{F})$, $\tilde{\boldsymbol{u}}=\mathrm{vec}(\boldsymbol{U})$, and $\tilde{\boldsymbol{v}}=\mathrm{vec}(\boldsymbol{V})$. We define matrix $\boldsymbol{O}=\alpha(\boldsymbol{S}^{\mathrm{T}}\boldsymbol{S}+\boldsymbol{E}_{m\times m}-\boldsymbol{S}-\boldsymbol{S}^{\mathrm{T}})+\boldsymbol{E}_{m\times m}$ and $\boldsymbol{\Theta}=\boldsymbol{E}_{l\times l}\otimes\boldsymbol{O}\in\mathbf{R}^{ml\times ml}$, $\otimes$ is Kronecker product and $\boldsymbol{E}$ is an identity matrix. This optimization problem (13) is a standard QP problem which can be solved by off-the-shelf QP tools. We adopt interior point methods to solve (13) by the quadprag function in Matlab.

Update $\overline{\boldsymbol{W}}$, with fixed $\boldsymbol{P}$ and $\boldsymbol{S}$. The linear model $f(\boldsymbol{x}_i,\boldsymbol{W})=\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i$ is utilized to predict the label of training examples. The problem (9) can be stated as follows:

$$\min_{\boldsymbol{W}}\sum_{i=1}^{m}\|\boldsymbol{W}^{\mathrm{T}}\boldsymbol{x}_i-\boldsymbol{p}_i\|_2^2+\lambda\|\boldsymbol{W}\|_{\mathrm{F}}^2. \tag{16}$$

This is an unconstrained optimization problem. The predictive performance can be improved by the kernel technology. Let $\Phi(\cdot)$: $\mathbf{R}^d\rightarrow\mathbf{R}^h$ be a feature mapping which maps the feature space to a Hilbert space with h-dimensions. The linear model is rewritten as $g(\boldsymbol{x}_i)=\overline{\boldsymbol{W}}^{\mathrm{T}}\Phi(\boldsymbol{x}_i)$. We convert the problem (16) into an equality constrained minimization as follows:

$$\min_{\boldsymbol{W}}\sum_{i=1}^{m}\|\boldsymbol{\xi}_i\|_2^2+\lambda\|\overline{\boldsymbol{W}}\|_{\mathrm{F}}^2,$$
$$\text{s.t.}\begin{cases}\overline{\boldsymbol{W}}^{\mathrm{T}}\Phi(\boldsymbol{x}_i)-\boldsymbol{P}_i=\boldsymbol{\xi}_i\\ 1\leqslant i\leqslant m\end{cases}. \tag{17}$$

The method of Lagrange multipliers can be used to solve this problem. The Lagrange function is stated as follows:

$$\mathcal{L}(\overline{\boldsymbol{W}},\boldsymbol{I},\boldsymbol{A})=\mathrm{tr}(\boldsymbol{I}^{\mathrm{T}}\boldsymbol{I})+\lambda\mathrm{tr}(\overline{\boldsymbol{W}}^{\mathrm{T}}\overline{\boldsymbol{W}})-\mathrm{tr}(\boldsymbol{A}^{\mathrm{T}}(\boldsymbol{\Phi}\overline{\boldsymbol{W}}-\boldsymbol{P}-\boldsymbol{I}))$$

where $\boldsymbol{\Phi}=[\Phi(\boldsymbol{x}_1),\Phi(\boldsymbol{x}_2),\cdots,\Phi(\boldsymbol{x}_m)]^{\mathrm{T}}$, $\boldsymbol{I}=[\boldsymbol{\xi}_1,\boldsymbol{\xi}_2,\cdots,\boldsymbol{\xi}_m]^{\mathrm{T}}$, and $\boldsymbol{A}=[\boldsymbol{a}_1,\boldsymbol{a}_2,\cdots,\boldsymbol{a}_m]^{\mathrm{T}}$ is the Lagrange multiplier matrix. The optimal conditions of (17) are

$$\nabla\mathcal{L}(\overline{\boldsymbol{W}},\boldsymbol{I},\boldsymbol{A})=\begin{bmatrix}\frac{\partial\mathcal{L}}{\partial\overline{\boldsymbol{W}}}\\ \frac{\partial\mathcal{L}}{\partial\boldsymbol{I}}\\ \frac{\partial\mathcal{L}}{\partial\boldsymbol{A}}\end{bmatrix}=\begin{bmatrix}2\lambda\overline{\boldsymbol{W}}-\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{A}\\ 2\boldsymbol{I}+\boldsymbol{A}\\ \boldsymbol{\Phi}\overline{\boldsymbol{W}}-\boldsymbol{P}-\boldsymbol{I}\end{bmatrix}=\boldsymbol{0}.$$

The optimal solution $\overline{\boldsymbol{W}}^*$ can be obtained as follows:

$$\begin{cases}\overline{\boldsymbol{W}}^*=\dfrac{\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{A}}{2\lambda}\\ \boldsymbol{A}=\left(\dfrac{1}{2\lambda}\boldsymbol{K}+\dfrac{1}{2}\boldsymbol{E}\right)^{-1}\boldsymbol{P}\end{cases} \tag{18}$$

where $\boldsymbol{K}=\boldsymbol{\Phi}\boldsymbol{\Phi}^{\mathrm{T}}$ is the kernel matrix with its element $k_{i,j}=\kappa(\boldsymbol{x}_i,\boldsymbol{x}_j)$, and $\kappa(\cdot,\cdot)$ is a kernel function. For PL-LCSA, we use Gaussian kernel to calculate $\boldsymbol{K}$. The modeling output matrix $\boldsymbol{F}$ can be calculated by $\boldsymbol{F}=\boldsymbol{\Phi}\overline{\boldsymbol{W}}^*=\boldsymbol{K}\boldsymbol{A}/2\lambda$. Unseen instance $\boldsymbol{x}$ can be predicted [11] by

$$\boldsymbol{y}^*=\arg\max_i\sum_{j=1}^{m}a_{j,i}\kappa(\boldsymbol{x},\boldsymbol{x}_j) \tag{19}$$

where $a_{i,j}$ is the element of matrix $\boldsymbol{A}$, and $\boldsymbol{y}^*$ is the predicted label for $\boldsymbol{x}$. The pseudo-code of PL-LCSA is summarized as Algorithm 1.

Algorithm 1 Pseudo-code of PL-LCSA

Input:
$\mathcal{D}$: training label set $\mathcal{D}=\{(\boldsymbol{x}_i,Z_i)|1\leqslant i\leqslant m\}$
$\boldsymbol{x}$: the unseen instance
Parameter:
k: the number of nearest neighbors
δ, α, β, γ, λ: the trade-off coefficients
T: the number of maximum iterations
Output:
$\boldsymbol{y}^*$: the predicted label of $\boldsymbol{x}$
Process:
1: Calculate label correlation matrix $\boldsymbol{B}$ according to (3)
2: Calculate $\boldsymbol{H}$ by (5)
3: Calculate kernel matrix $\boldsymbol{K}$
4: Initialize $\boldsymbol{S}$ according to (10)
5: Initialize $\boldsymbol{P}$ according to (11)
6: **Repeat**
7: Update matrix $\boldsymbol{A}$ according to (18)
8: Update modeling output matrix $\boldsymbol{F}=\boldsymbol{K}\boldsymbol{A}/2\lambda$
9: Update matrix $\boldsymbol{S}$ according to (13)
10: Update matrix $\boldsymbol{P}$ by solving problem (15)
11: **Until** convergence or the maximum number of iterations
12: **Return** $\boldsymbol{y}^*$ according to (19)

4. Experiments

4.1 Experimental setup

In this subsection, two series of comparative experiments based on synthetic data sets and real-world data sets are conducted to evaluate the performance of PL-LCSA. Table 1 summarizes the characteristics of six multi-class

data sets. Following the widely-used controlling protocol [6,8,11,14,25−27], artificial data sets are generated from multi-class data sets by three controlling parameters p, ϵ, and r. Here, p is the proportion of example which is ambiguous. r is the number of the false positive labels in the candidate label set (i.e., $|Z_i|=r+1$). ϵ is the probability that a false positive label co-occurs with the ground-truth label. The choice of false positive label is important, because it is the key factor of the label relationship in the synthetic data sets. In this paper, the reason why we choose multi-label data sets as synthetic data sets is that the multi-label data can be used to generate label correlation information. Firstly, (3) is used to calculate label correlation in multi-label data sets. Then the most relevant label corresponding to the ground-truth label is taken as the false positive label. Table 2 summarizes the characteristics of six real-world data sets, where Avg. CLs is the average number of candidate labels.

Table 1 Characteristics of the multi-class data sets

Data set	Instance	Feature	PCA	Class
Enron	196	623	−	7
Medical	752	1 287	−	30
Scene	2 230	294	−	6
Bibtex	2 744	1 894	800	89
Mediamill	2 854	120	−	10
Tmc2007	3 130	15 418	1 000	21

Table 2 Characteristics of real-world partial label data sets

Data set	Example	Feature	Class	Avg. CLs	Task domain
FG-NET	1 002	262	78	7.48	Facial age estimation [28]
Lost	1 122	108	16	2.23	Automatic face naming [6]
MSRCv2	1 758	48	23	3.16	Object classification [1]
BirdSong	4 998	38	13	2.18	Bird song classification [29]
Soccer Player	17 472	279	171	2.09	Automatic face naming [3]
Yahoo! News	22 991	163	219	1.91	Automatic face naming [30]

The data sets in Tabel 1 are derived from multi-label benchmark data set by retaining instances with only one relevant label. They can be collected from Mulan (http://mulan.sourceforge.net/index.html). Because the features of Bibtex and Tmc2007 are relatively sparse, we make dimensionality reduction by principal component analysis (PCA), and the feature dimensions after dimensionality reduction are shown in the fourth column of Table 1.

In this paper, five partial label learning algorithms are utilized for comparative studies. Each algorithmis configured with the following literature:

(i) PL-KNN [5]: a k-nearest neighbor approach which makes the prediction for unseen instances by averaging the labeling information of its k-nearest neighbor (suggested configuration: k=10).

(ii) IPAL[8]: an instance-based approach which disambiguates the candidate label set via an iterative label propagation procedure (suggested configuration: k =10 and α=0.95).

(iii) PL-ECOC[21]: a disambiguation-free approach which learns the predictive model by the error-correcting output codes (ECOC) technique (suggested configuration: $\tau=0.1\cdot|D|$, $L=\lceil 40\cdot\log_2(l)\rceil$).

(iv) PL-AGGD [11]: an approach which disambiguates the candidate label sets by adaptive graph to overcome the noise and outliers in the feature space (suggested configuration: k=10, λ=1, μ=1, γ=0.05, and T=10).

(v) SDIM [14]: an approach which aims to maximize the latent semantic differences of the two instances whose ground-truth labels are definitely different (suggested configuration: λ=0.05, β=0.001).

The parameters of PL-LCSA are set as δ=0.5, α=0.5, β=0.05, γ=0.5, λ=1, k=10, and T=10. Ten-fold cross-validation is executed in each algorithm, and mean prediction accuracy and standard deviations will be recorded.

4.2 Results and discussion

4.2.1 Real-world data sets

Table 3 is the summary classification accuracy (mean± standard deviation (std)) of each comparing algorithm on the real-world data sets. • indicates the PL-LCSA is statistically superior/inferior to the comparing algorithms on each data set (pairwise t-test at 0.05 significance level). The performance of each algorithm is poor on the face and gesture recognition network (FG-NET) aging data set, because its Avg. CLs is extremely large.

As shown in Table 3, it can be seen that:

(i) On the Lost, MSRCv2 and Soccer Player data sets, the performance of PL-LCSA is superior to all the comparing algorithms;

(ii) On the BirdSong data set, the performance of PL-LCSA is comparable to the PL-ECOC and superior to the other comparing algorithms;

(iii) On the FG-NET data set, the performance of PL-

LCSA is comparable to the SDIM, PL-AGGD, PL-KNN and superior to the PL-ECOC and IPAL.

(iv) PL-LCSA is never outperformed by any comparing algorithms.

Table 3 Classification accuracy (mean±std) of each comparing algorithm on real-world partial label data sets

Data set	PL-LCSA	SDIM	PL-AGGD	PL-ECOC	IPAL	PL-KNN
Lost	0.789±0.030	0.797±0.030	0.744±0.020•	0.706±0.043•	0.645±0.034•	0.459±0.039•
MSRCv2	0.562±0.021	0.500±0.023•	0.509±0.028•	0.427±0.024•	0.531±0.037•	0.418±0.046•
BirdSong	0.756±0.008	0.734±0.012•	0.734±0.009•	0.751±0.013	0.712±0.015•	0.603±0.013•
Soccer Player	0.596±0.013	0.577±0.016•	0.539±0.016•	0.169±0.005•	0.544±0.014•	0.494±0.012•
Yahoo! News	0.670±0.008	0.663±0.013	0.647±0.009•	0.561±0.011•	0.607±0.012•	0.471±0.005•
FG-NET	0.079±0.020	0.076±0.037	0.076±0.027	0.005±0.007•	0.061±0.018•	0.066±0.018

From Table 3, PL-LCSA is capable of better performance compared with other comparison methods in the real-world data sets. Because three items are integrated in the PL-LCSA, we set three sets of comparative experiments in real-world data sets to determine the effect of these modules by parameters δ and γ. The first set of comparative experiments shows the performance of the adaptive graph. The second set shows the performance of the adaptive graph and the label correlation. The third set of comparative experiments shows the performance of PL-LCSA.

Table 4 shows the classification accuracy (mean±std) of these three sets of comparative experiments. A significantly increased performance was observed in the second set experiment compared with the first set experiment on MSRCv2, BirdSong, Soccer Player, and Yahoo! News. It is because the label correlation has contributed to the partial label learning. The performance of the third set of experiment achieves the better performance to the second, and the reason is that the label correlation and the semantic difference maximization can jointly promote the performance.

Table 4 Classification accuracy (mean±std) of control variables for PL-LCSA on real-world data sets

Data set	$\delta=0.5,\gamma=0.5$	$\delta=0.5,\gamma=0$	$\delta=0,\gamma=0$
Lost	0.789±0.030	0.767±0.032	0.746±0.025
MSRCv2	0.562±0.021	0.544±0.030	0.507±0.028
BirdSong	0.756±0.008	0.751±0.010	0.733±0.011
Soccer Player	0.596±0.013	0.589±0.013	0.538±0.015
Yahoo! News	0.670±0.008	0.666±0.009	0.648±0.009
FG-NET	0.079±0.020	0.071±0.023	0.077±0.024

4.2.2 Synthetic data sets

Fig. 1 illustrates the classification accuracy of each comparing algorithm on the synthetic data sets as the co-occurring probability ϵ varies from 0.1 to 0.9 with step size 0.1, where r=2 and p=1.

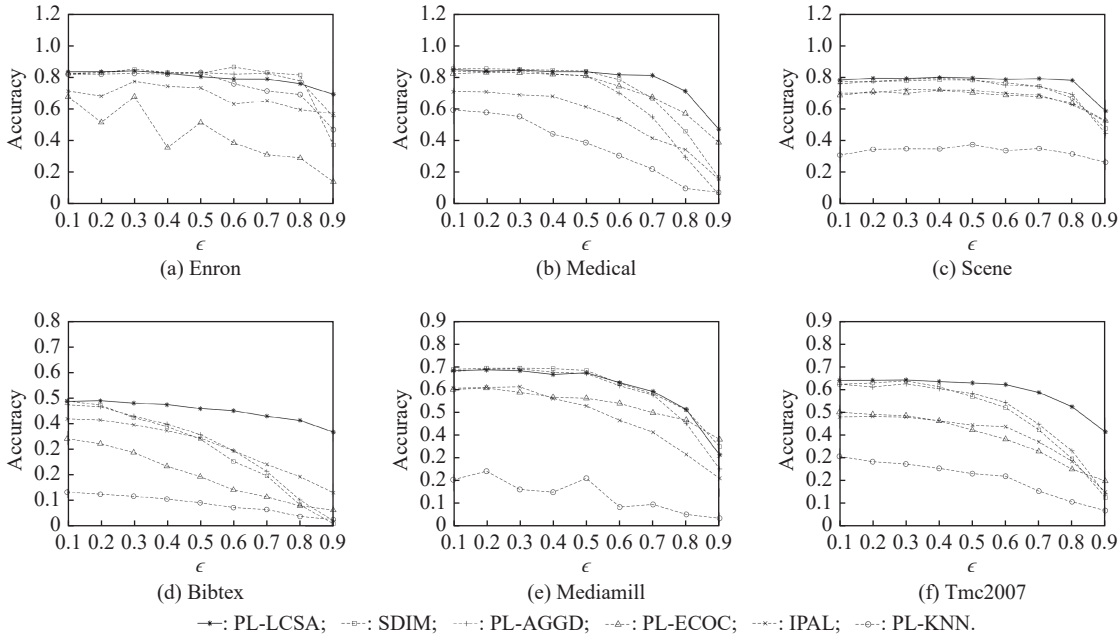
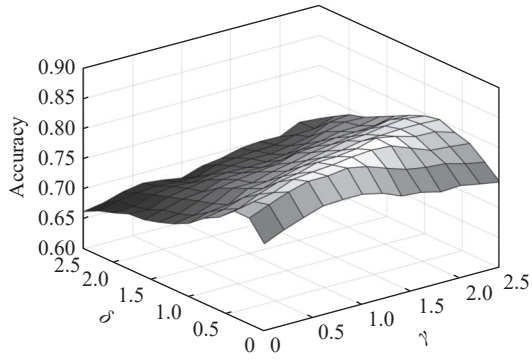


Fig. 1 Classification performance on synthetic data sets with ϵ ranging from 0.1 to 0.9 with step size 0.1 (r=2, p=1)

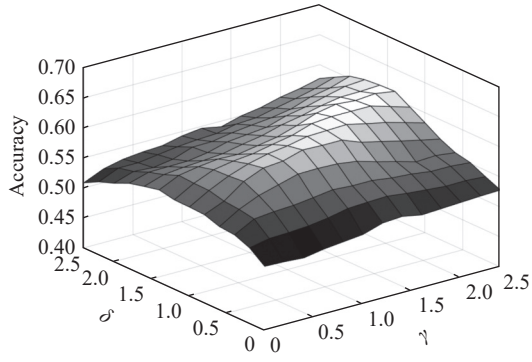
In general, PL-LCSA achieves competitive or better performance at synthetic data sets. Fig. 1 reveals that there has been a gradual decrease in the second half. The reason may be that the greater the similarity between the ground-truth label and the false positive label is, the more difficult it is to distinguish. Compared with the PL-AGGD, PL-LCSA achieves competitive performance at a smaller value of ϵ and better performance at a bigger value of ϵ on Enron, Scene, and Mediamill. The reason is that the size of label space on Enron, scene, and Mediamill is small, and the synthetic data does not have the information about the label correlation at a smaller value of ϵ and it is disadvantageous to PL-LCSA. On the contrary, when the value of ϵ is high, PL-LCSA achieves superior performance against PL-AGGD. It shows that the more the information about the label correlation is, the better the performance of PL-LCSA is.

4.2.3 Parameter sensitivity

Fig. 2 shows the accuracy of PL-LCSA under different configurations for parameters δ and γ on Lost and MSRCv2. As γ increases, PL-LCSA starts to take into consideration the semantic difference maximization criterion and the classification accuracy increases. For δ, as the weight of the label correlation increases, the classification accuracy decreases first, and then increases. In practice, we suggest users to choose δ and γ around 0.5 for all data sets.



(a) Varying δ and γ on lost



(b) Varying δ and γ on MSRCv2

Fig. 2 Parameter sensitivity analysis for PL-LCSA

5. Conclusions

In this paper, we propose a unified framework that simultaneously focuses on the label and feature space. Meanwhile, this work generates fresh insight into the acquisition of the learning information from the label space, i.e., the label correlation. The framework integrates the label correlation, the adaptive graph, and the semantic difference maximization criterion. The relationship of instances can be learned by the adaptive graph in the feature space, the semantic difference analyzes the label relationship at the instance level, and the label correlation learns at the global label level. An effective optimization method is also proposed for this framework. Experiments on real-world and artificial data sets have demonstrated the superiority of PL-LCSA to the state-of-the-art partial label learning approaches.

References

[1] LIU L P, DIETTERICH T G. A conditional multinomial mixture model for superset label learning. Advances in Neural Information Processing Systems, 2012, 1: 548–556.
[2] ZHOU D, ZHANG Z, ZHANG M L, et al. Weakly supervised POS tagging without disambiguation. ACM Transactions on Asian and Low-Resource Language Information Processing, 2018, 17(4): 1–19.
[3] ZENG Z N, XIAO S J, JIA K, et al. Learning by associating ambiguously labeled images. Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, 2013: 708–715.
[4] YAO Y, DENG J H, CHEN X H, et al. Deep discriminative CNN with temporal ensembling for ambiguously-labeled image classification. Proc. of the AAAI Conference on Artificial Intelligence, 2020, 34(7): 12669–12676.
[5] HULLERMEIER E, BERINGER J. Learning from ambiguously labeled examples. Intelligent Data Analysis, 2006, 10(5): 419–439.
[6] COUR T, SAPP B, TASKAR B. Learning from partial labels. Journal of Machine Learning Research, 2011, 12: 1501–1536.
[7] NGUYEN N, CARUANA R. Classification with partial labels. Proc. of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2008: 551–559.
[8] ZHANG M L, YU F. Solving the partial label learning problem: an instance-based approach. Proc. of the International Joint Conference on Artificial Intelligence, 2015: 4048–4054.
[9] ZHANG M L, ZHOU B B, LIU X Y. Partial label learning via feature-aware disambiguation. Proc. of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016: 1335–1344.
[10] GONG C, LIU T L, TANG Y Y, et al. A regularization approach for instance-based superset label learning. IEEE Trans. on Cybernetics, 2017, 48(3): 967–978.
[11] WANG D B, LI L, ZHANG M L. Adaptive graph guided disambiguation for partial label learning. Proc. of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2019: 83–91.
[12] LYU G Y, FENG S H, WANG T, et al. GM-PLL: graph

matching based partial label learning. IEEE Trans. on Knowledge and Data Engineering, 2021, 33(2): 521–535.
[13] ZHU Y, KWOK J T, ZHOU Z H. Multi-label learning with global and local label correlation. IEEE Trans. on Knowledge and Data Engineering, 2017, 30(6): 1081–1094.
[14] FENG L, AN B. Partial label learning by semantic difference maximization. Proc. of the International Joint Conference on Artificial Intelligence, 2019: 2294–2300.
[15] JIN R, GHAHRAMANI Z. Learning with multiple labels. Proc. of the Annual Conference on Neural Information Processing Systems, 2002, 2: 897–904.
[16] YU F, ZHANG M L. Maximum margin partial label learning. Proc. of the Asian Conference on Machine Learning, 2016: 96–111.
[17] CHEN Y C, PATEL V M, CHELLAPPA R, et al. Ambiguously labeled learning using dictionaries. IEEE Trans. on Information Forensics and Security, 2014, 9(12): 2076–2088.
[18] TANG C Z, ZHANG M L. Confidence-rated discriminative partial label learning. Proc. of the AAAI Conference on Artificial Intelligence, 2017, 31(1): 2611–1618.
[19] LYU G Y, FENG S H, LI Y D, et al. Hera: partial label learning by combining heterogeneous loss with sparse and low-rank regularization. ACM Trans. on Intelligent Systems and Technology, 2020, 11(3): 1–19.
[20] ZHANG Y B, YANG G, ZHAO S Y, et al. Partial label learning via generative adversarial nets. Proc. of the European Conference on Artificial Intelligence, 2020: 1674–1681.
[21] ZHANG M L, YU F, TANG C Z. Disambiguation-free partial label learning. IEEE Trans. on Knowledge and Data Engineering, 2017, 29(10): 2155–2167.
[22] WU X, ZHANG M L. Towards enabling binary decomposition for partial label learning. Proc. of the International Joint Conference on Artificial Intelligence, 2018: 2868–2874.
[23] YUILLE A L, RANGARAJAN A. The concave-convex procedure (CCCP). Advances in Neural Information Processing Systems, 2002, 2: 1033–1040.
[24] SRIPERUMBUDUR B K, LANCKRIET G R G. On the convergence of the concave-convex procedure. Proc. of the Annual Conference on Neural Information Processing Systems, 2009, 9: 1759–1767.
[25] XU S, YANG M, ZHOU Y, et al. Partial label metric learning by collapsing classes. International Journal of Machine Learning and Cybernetics, 2020, 11: 2453–2460.
[26] LYU J Q, XU M, FENG L, et al. Progressive identification of true labels for partial-label learning. Proc. of the International Conference on Machine Learning, 2020: 6500–6510.
[27] YAN Y, GUO Y H. Partial label learning with batch label correction. Proc. of the AAAI Conference on Artificial Intelligence, 2020, 34(4): 6575–6582.
[28] PANIS G, LANITIS A, TSAPATSOULIS N, et al. Overview of research on facial ageing using the FG-NET ageing database. IET Biometrics, 2016, 5(2): 37–46.
[29] BRIGGS F, FERN X Z, RAICH R. Rank-loss support instance machines for MIML instance annotation. Proc. of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012: 534–542.
[30] GUILLAUMIN M, VERBEEK J, SCHMID C. Multiple instance metric learning from automatically labeled bags of faces. European Conference on Computer Vision, 2010: 634–647.

Biographies

GE Lingchi was born in 1997. He received his B.S. degree from Chang'an University, Xi 'an, China, in 2018. He is currently pursuing his M.S. degree in computer science at Xidian University, Xi'an, China. His research interests are pattern recognition, partial label learning, and multi-label learning.
E-mail: 15109285306@163.com

FANG Min was born in 1965. She received her B.S. degree in computer control, M.S. degree in computer software engineering, and Ph.D. degree in computer application from Xidian University, Xi'an, China, in 1986, 1991, and 2004, respectively, where she is currently a professor. Her research interests include intelligent information process, multi-agent system, and network technology.
E-mail: fanglabtg@163.com

LI Haikun was born in 1990. He received his M.S. degree from Yunnan University, Kunming, China, in 2017. He is currently a Ph.D. candidate in technology of computer application at Xidian University, Xi'an, China. His research interests include pattern recognition, machine learning, and partial-label learning.
E-mail: haikun1990@163.com

CHEN Bo was born in 1993. He received his B.S. degree in 2014 from Xi'an University of Post & Telecomunications, Xi 'an, China. He is currently a Ph.D. candidate in computer science at Xidian University, Xi'an, China. His research interests include pattern recognition, machine learning, and time series.
E-mail: bchen_0314@stu.xidian.edu.cn