Partial Label Learning Tailored Graph Construction

Fuchao Yang
College of Software Engineering, Southeast University
Nanjing, China
yangfc@seu.edu.cn

Yongqiang Dong
School of Computer Science and Engineering, Southeast University
Nanjing, China
dongyq@seu.edu.cn

Yuheng Jia*
School of Computer Science and Engineering, Southeast University
Nanjing, China
yhjia@seu.edu.cn

***Abstract*—In partial label learning (PLL), each sample is annotated with a group of candidate labels, among which only one label is correct. The key of PLL is to find the ground-truth label concealed in the candidate label set, which is known as label disambiguation. The instance relationships captured by a graph play a central role in label disambiguation, as if two samples are close to each other in the feature space, they are expected to share the ground-truth same label. However, the existing PLL methods simply use the feature matrix to construct the graph without considering the characteristics of PLL. In this paper, we propose a novel graph construction model that is tailored to PLL. Specifically, we first build a local similarity matrix by reconstructing a sample by its neighbors. Second, we design a dissimilarity matrix to specify the highly dissimilar samples according to the available partial labels, and further enhance it by dissimilarity propagation. As the similarity and the dissimilarity matrices form an adversarial relationship, the enhanced dissimilarity matrix is used to refine the similarity matrix. Then, the proposed model is finally formulated as a dissimilarity propagation guided graph learning problem, which is solved by the inexact augmented Lagrange multiplier method. Extensive experiments on artificial as well as real-world partial label data sets demonstrate that the learned graph can correctly capture the similarity relationships among samples and improve the classification performance of different graph-based PLL methods. The code implementation is publicly available at https://github.com/Yangfc-ML/PL-TGC.**

***Index Terms*—weakly supervised learning, partial label learning, graph learning.**

I. Introduction

Partial label learning (PLL) [1]–[6] is an important weakly supervised learning framework. In PLL, each training example is annotated with a set of candidate labels, among which only one label is valid. Compared to traditional supervised learning, PLL avoids precisely annotating ground-truth labels on large-scale data sets, which greatly reduces the labeling cost. Due to this advantage, PLL has been applied to many real-world scenarios such as automatic image annotation [7], [8], web mining [9], and ecoinformatics [10].

Formally, let $\mathcal{X} = \mathbb{R}^d$ be the d-dimensional feature space and $\mathcal{Y} = \mathbb{R}^q$ be the label space with q labels. Given the partial label training set $\mathcal{D} = \{(x_i, S_i) \mid 1 \leq i \leq m\}$, where $x_i \in \mathcal{X}$ is a d-dimensional feature vector and $S_i \in \mathcal{Y}$ is the associated candidate label set. PLL aims to induce a multi-class classifier $f : \mathcal{X} \to \mathcal{Y}$ from $\mathcal{D}$. However, as the ground-truth label of a sample (e.g., x_i) is concealed in its candidate label set S_i which is not directly available, inducing a classifier in PLL is very challenging.

The basic strategy to solve PLL is label disambiguation, i.e., identify the ground-truth label of samples from their candidate labels. According to the disambiguation strategies, existing methods can be roughly divided into two categories, i.e., the average-based strategy and the identification-based strategy. The average-based strategy assumes that each candidate label contributes equally to the model training and makes predictions by averaging their modeling outputs [7], [11], [12]. The identification-based strategy considers the ground-truth label as a latent variable and identifies it through an iterative refining procedure [13]–[15].

Recently, graph-based disambiguation methods become popular in PLL [16], [17]. These methods usually first construct a graph in the feature space to measure the similarities among samples. Then, they use label propagation or graph Laplacian regularization to achieve label disambiguation, as if two samples are close to each other in the feature space, they are expected to share the same label. Apparently, the quality of the constructed graph is important, however, those methods do not pay much attention to graph construction. Specifically, they only use the feature matrix to build the graph by an existing approach like the k-nearest neighbor graph, which ignores the valuable supervision information in PLL.

Realizing the importance of the constructed graph in PLL, in this paper, we propose a novel graph learning method that is tailored to PLL named PL-TGC (Partial Label Learning **T**ailored **G**raph **C**onstruction), which simultaneously takes the information of feature space and label space into account. Specifically, we first reconstruct each sample by its top k-nearest neighbors, and use the reconstruction coefficient as the weight of the similarity matrix. Then, we extract a dissimilarity matrix through the partial labels, i.e., if two samples do not share any common candidate labels, they must belong to different classes and the dissimilarity between them is very large. As the initial dissimilarity matrix is sparse, we further propagate it by local consistency, i.e., if two samples are similar to each other, they are expected to have the similar dissimilarity codings. Being aware of the adversarial relationship between the similarity matrix and the dissimilarity matrix, i.e., a larger (resp. smaller) similarity between two samples means a smaller (resp. larger) dissimilarity between them, we use the enhanced the dissimilarity matrix to refine the similarity matrix. Finally,

*Corresponding author: yhjia@seu.edu.cn

the proposed model is formulated as an adversarial learning regularized self-representation model, which is optimized by inexact augmented Lagrange multipliers (IALMs). Extensive experiments on artificial and real-world partial label data sets demonstrate the effectiveness of the proposed method on graph construction, and the existing graph-based PLL algorithms can be significantly improved by incorporating PL-TGC.

II. The Proposed Approach

Following the notations in the introduction, $X = [x_1, x_2, ..., x_m]^T \in \mathbb{R}^{m\times d}$ denotes the feature matrix with m and d being the number of samples and the dimension of features, $Y = [y_1, y_2, ..., y_m]^T \in \{0,1\}^{m\times q}$ represents the corresponding label matrix with q being the number of labels, where y_{ij}=1 means that the j-th label is a candidate label of sample x_i, and y_{ij}=0 represents that it is not the candidate label. Following [18], we assume each sample can be linearly reconstructed by its neighboring samples, i.e.,

$$\begin{aligned}&\min_{W}\sum_{j=1}^{m}\left\|x_j-\sum_{(x_i,x_j)\in\mathcal{N}}W_{ij}\cdot x_i\right\|_2^2\\&\text{s.t. } W^{\mathsf{T}}1_m=1_m, 0\le W_{ij}\le 1, W_{ij}=0, \text{if } (x_i,x_j)\notin\mathcal{N},\end{aligned}\tag{1}$$

where $(x_i,\ x_j) \in \mathcal{N}$ means that x_i belongs to x_j's top k-nearest neighbors. $1_m \in \mathbb{R}^{m\times 1}$ is an all ones vector, and $||\cdot||_2$ represents the l_2 norm. $W \in \mathbb{R}^{m\times m}$ is the reconstruction coefficient matrix, and $W_{ij} = 0, \text{if } (x_i, x_j) \notin \mathcal{N}$ ensures that a sample is only represented by its local neighbors. By solving Eq. (1), W can reconstruct each sample by its nearest neighbors, which could reveal the local geometric structures of the data set. And accordingly, it can act as the weight matrix of a graph to capture the similarity relationships of instances. $W^T 1_m = 1_m$ and $0 \le W \le 1$ make W a normalized non-negative graph weight matrix.

Eq. (1) constructs the similarity matrix by exploring the feature space. To further exploit the valuable information in PLL, we use the candidate labels to construct a dissimilarity matrix $D_0 \in \mathbb{R}^{m\times m}$. Specifically, for each sample, $y_i = [y_{1i}, y_{2i}, ..., y_{qi}]^T$ is the label vector of x_i, and $y_{ij} = 1$ means the j-th label lies in the candidate label set of x_i, otherwise $y_{ij} = 0$. If two samples share fewer common candidate labels, they are less likely to belong to the same class. Particularly, if no common candidate labels exist between two samples, they definitely belong to the different classes. Based on this, the dissimilarity matrix is developed as:

$$\begin{aligned}&D_{0ij}=\begin{cases}T_{ij}, \text{ if } (x_i,x_j)\in\mathcal{N} \text{ and } T_{ij}\ge \text{threshold}\\ 0\ , \text{otherwise}\end{cases}\\&\forall\ i,j, 0\le i,j\le m,\end{aligned}\tag{2}$$

where $T_{ij} = 1 - \Phi(y_i^T y_j)$, $\Phi(\cdot)$ is a normalization operator, i.e., $\Phi(y_i^T y_j) = (y_i^T y_j - \min(Y_j^T y_j))/(\max(Y_j^T y_j) - \min(Y_j^T y_j))$ and $Y_j \in \mathbb{R}^{k\times q}$ contains all label vectors of x_j's top k-nearest neighbors. This operator guarantees that D_0 lies in the range of [0,1]. Moreover, we remove the small values of D_0 by a predefined threshold, hoping the dissimilarity relations in D_0 are reliable. As D_0 is extracted from the label space, which indicates the dissimilarity relationships of samples, while W reveals the similarity relationships among samples from the feature space, W and D_0 form an adversarial relationship, i.e., a larger (resp. smaller) element in W implies a smaller (resp. larger) element in D_0. Therefore, this adversarial relationship is formulated as

$$\min_{W,D_0} \|D_0 \odot W\|_F^2,\tag{3}$$

where $\odot$ is the elementwise product of matrices, $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. As the non-negative elements of W only exist locally, in Eq. (2), we can also construct a local dissimilarity matrix.

Nevertheless, D_0 is a quite sparse matrix with limited non-zero elements. To strengthen the effect of the dissimilarity relationships, we propose to propagate the limited positive elements of D_0 to produce a denser dissimilarity matrix $D \in \mathbb{R}^{m\times m}$. Specifically, D_{ij} should equal to D_{0ij} if $D_{0ij} > 0$, which means the highly reliable dissimilarity relationships in D_0 are retained in D. Moreover, each element of D should lie in the range of [0,1] to make it a well-defined dissimilarity measure. Finally, the i-th column of D (i.e., $D_{\cdot i}$) could represent the dissimilarity relationship of x_i to other samples, and $D_{\cdot i}$ should be similar to $D_{\cdot j}$ if x_i is close to x_j. Therefore, the dissimilarity propagation of D could be formulated as

$$\begin{aligned}&\min_{D}\sum_{i,j=1}^{m}\|D_{\cdot i}-D_{\cdot j}\|_2^2\cdot W_{ij}\\&\text{s.t. } \forall\ i,j, 0\le D_{ij}\le 1, D_{ij}=D_{0ij}, \text{ if } D_{0ij}\ne 0.\end{aligned}\tag{4}$$

After getting an enhanced dissimilarity matrix D, we could use it to further refine the similarity matrix W. Taking all the above considerations into account, the proposed model finally becomes:

$$\begin{aligned}&\min_{W,D}\sum_{j=1}^{m}\left\|x_j-\sum_{(x_i,x_j)\in\mathcal{N}}W_{ij}\cdot x_i\right\|_2^2+\lambda_1\|D\odot W\|_F^2+\lambda_2 Tr(DLD^{\mathsf{T}})\\&\text{s.t. } W^{\mathsf{T}}1_m=1_m, \forall\ i,j, 0\le W_{ij}, D_{ij}\le 1,\ D_{ij}=D_{0ij}, \text{ if } D_{0ij}\ne 0,\end{aligned}\tag{5}$$

where $L \in \mathbb{R}^{m\times m} = D_S - \frac{W^T+W}{2}$ is a graph Laplacian matrix, and $D_S \in \mathbb{R}^{m\times m}$ is a diagonal matrix with the i-th diagonal element being $\sum_{j=1}^{m}(W_{ij} + W_{ji})/2$. λ_1, $\lambda_2 \ge 0$ are two hyper-parameters to balance different terms. When Eq. (5) is solved, the enhanced dissimilarity matrix is generated by propagating the limited dissimilarity relationships in the label space, and further refines the similarity matrix generated from the feature space by a novel adversarial term. Therefore, we can expect the similarity graph captured by W is more informative than that used in the previous graph-based PLL methods.

III. Optimization

We adopt the inexact augmented Lagrange multipliers (IALMs) to solve the problem in Eq. (5). Introducing two

auxiliary matrices $A \in \mathbb{R}^{m\times m}$ and $B \in \mathbb{R}^{m\times m}$, Eq. (5) is equivalently written as

$$\begin{aligned}&\min_{A,B,W,D} \sum_{j=1}^{m} \left\| x_j - \sum_{(x_i,x_j)\in\mathcal{N}} W_{ij}\cdot x_i \right\|_2^2 + \lambda_1 \|A\odot B\|_{\mathrm{F}}^2 + \lambda_2 Tr(DLD^{\mathsf{T}}) \\ &\text{s.t. } W^{\mathsf{T}}1_m = 1_m, \forall\, i,j, 0 \le A_{ij}, B_{ij} \le 1, D = A, W = B, \\ &\quad A_{ij} = D_{0ij}, \text{ if } D_{0ij} \neq 0.\end{aligned} \tag{6}$$

The augmented Lagrangian form of Eq. (6) is given by

$$\begin{aligned}&\arg\min_{A,B,W,D} \sum_{j=1}^{m} \left\| x_j - \sum_{(x_i,x_j)\in\mathcal{N}} W_{ij}\cdot x_i \right\|_2^2 + \lambda_1 \|A\odot B\|_{\mathrm{F}}^2 + \lambda_2 Tr(DLD^{\mathsf{T}}) \\ &\qquad + \langle Y_1, D-A\rangle + \frac{\mu}{2}\|D-A\|_{\mathrm{F}}^2 + \langle Y_2, W-B\rangle + \frac{\mu}{2}\|W-B\|_{\mathrm{F}}^2 \\ &\text{s.t. } W^{\mathsf{T}}1_m = 1_m, \forall\, i,j, 0 \le A_{ij}, B_{ij} \le 1, A_{ij} = D_{0ij}, \text{ if } D_{0ij} \neq 0,\end{aligned} \tag{7}$$

where $Y_1 \in \mathbb{R}^{m\times m}$ and $Y_2 \in \mathbb{R}^{m\times m}$ are the Lagrange multipliers, $\mu \ge 0$ is a penalty parameter, and $\langle\cdot,\cdot\rangle$ returns the inner product of two matrices. To solve Eq. (7), the IALMs iteratively solves the following subproblems.

1) D subproblem is written as

$$\min_{D} \lambda_2 Tr(DLD^{\mathsf{T}}) + \frac{\mu}{2}\left\| D - A + \frac{Y_1}{\mu}\right\|_{\mathrm{F}}^2. \tag{8}$$

Eq. (8) reaches the minimum when its first-order derivative with respect to D vanishes, leading to

$$D = (\mu A - Y_1)(2\lambda_2 L + \mu I_{m\times m})^{-1}, \tag{9}$$

where $I_{m\times m} \in \mathbb{R}^{m\times m}$ is an identity matrix.

2) A subproblem is formulated as

$$\begin{aligned}&\min_{A} \lambda_1 \|A\odot B\|_{\mathrm{F}}^2 + \frac{\mu}{2}\left\| D - A + \frac{Y_1}{\mu}\right\|_{\mathrm{F}}^2 \\ &\text{s.t. } \forall\, i,j, 0 \le A_{ij} \le 1, A_{ij} = D_{0ij}, \text{ if } D_{0ij} \neq 0\end{aligned} \tag{10}$$

The closed-form solution of Eq. (10) is given by

$$\begin{aligned}&A = \Gamma_1\left(\Gamma_0\left(\frac{\mu D + Y_1}{2\lambda_1\langle B,B\rangle + \mu 1_{m\times m}}\right)\right) \\ &\text{s.t. } \forall\, i,j, A_{ij} = D_{0ij}, \text{ if } D_{0ij} \neq 0,\end{aligned} \tag{11}$$

where $1_{m\times m}$ is an $m\times m$ all ones matrix. Γ_0 and Γ_1 are thresholding operators in elementwise, i.e., $\Gamma_0(a) := \max(0,a), \Gamma_1(a) := \min(1,a)$.

3) W subproblem is represented as

$$\begin{aligned}&\min_{W} \sum_{j=1}^{m} \left\| x_j - \sum_{(x_i,x_j)\in\mathcal{N}} W_{ij}\cdot x_i \right\|_2^2 + \lambda_2 Tr(DLD^{\mathsf{T}}) \\ &\qquad + \frac{\mu}{2}\left\| W - B + \frac{Y_2}{\mu}\right\|_F^2 \\ &\text{s.t } W^{\mathsf{T}}1_m = 1_m.\end{aligned} \tag{12}$$

Eq. (12) can be separated column-wisely, and in each column, we have

$$\begin{aligned}&\min_{W_{\cdot j}} W_{\cdot j}^{\mathsf{T}}(G^j + \frac{\mu}{2} I_{m\times m)})W_{\cdot j} + (\lambda_2 C_{\cdot j}^{\mathsf{T}} - \mu B_{\cdot j}^{\mathsf{T}} + Y_{2\cdot j}^{\mathsf{T}})W_{\cdot j} \\ &\text{s.t. } W_{\cdot j}^{\mathsf{T}}1_m = 1,\end{aligned} \tag{13}$$

where $G^j = O^{x_j}(O^{x_j})^{\mathsf{T}} \in \mathbb{R}^{k\times k}$ is the local Gram matrix for x_j, i.e., $O^{x_j} = [x_j - x_{N_{j(1)}}, x_j - x_{N_{j(2)}}, ..., x_j - x_{N_{j(k)}}]^{\mathsf{T}} \in \mathbb{R}^{k\times d}$, $N_{j(i)}$ indicates x_i is one of x_j's top k-nearest neighbors. $C_{\cdot j} = [C_{1j}, C_{2j}, ..., C_{kj}]^{\mathsf{T}} \in \mathbb{R}^{k\times 1}$, and $C_{ij} = \|D_{\cdot i} - D_{\cdot j}\|_2^2$.

Algorithm 1 Numerical Solution of Eq. (5)

Input: PL training set $\mathcal{D}$, λ_1, λ_2, threshold, k.
Output: graph weight matrix.

1: Construct the weight matrix W_0 according to Eq. (1) and the dissimilarity matrix D_0 according to Eq. (2)
2: Initialize $W = D = A = B = Y_1 = Y_2 = 0_{m\times m}$
3: **while** not converged **do**
4: Update D by Eq. (9)
5: Update A by Eq. (11)
6: Update W by Eq. (13)
7: Update B by Eq. (15)
8: Update Y_1, Y_2, μ by Eq. (16)
9: check the convergence conditions $\|W-B\|_\infty < 10^{-8}$ and $\|D-A\|_\infty < 10^{-8}$
10: **end while**

As Eq. (13) is a standard quadratic programming (QP) problem, it can be solved by any QP tools.

4) B subproblem is expressed as

$$\begin{aligned}&\min_{B} \lambda_1 \|A\odot B\|_{\mathrm{F}}^2 + \frac{\mu}{2}\left\| W - B + \frac{Y_2}{\mu}\right\|_{\mathrm{F}}^2 \\ &\text{s.t. } \forall\, i,j, 0 \le B_{ij} \le 1.\end{aligned} \tag{14}$$

The analytical solution of Eq. (14) is given by

$$B = \Gamma_1\left(\Gamma_0\left(\frac{\mu W + Y_2}{2\lambda_1\langle A,A\rangle + \mu 1_{m\times m}}\right)\right). \tag{15}$$

Finally, the Lagrangian multiplier matrices and μ are updated by

$$\begin{cases} Y_1 \leftarrow Y_1 + D - A \\ Y_2 \leftarrow Y_2 + W - B \\ \mu = \min(1.1\mu, \mu_{\max}), \end{cases} \tag{16}$$

where $\mu_{\max}$=10^{10} is a predefined upper bound for μ.

Algorithm 1 summarizes the overall pseudo code, where it stops when the residuals of the optimized variables are less than 10^{-8}, i.e., $\|W-B\|_\infty < 10^{-8}$ and $\|D-A\|_\infty < 10^{-8}$, where $\|\cdot\|_\infty$ denotes the infinity norm of a matrix.

A. Computational Complexity Analysis

The computational complexity of Algorithm 1 is decided by steps 4-7. Specifically, step 4 involves the inversion of an $m\times m$ sparse matrix with the complexity of $\mathcal{O}(m^{2.373})$. Steps 5 and 7 can be efficiently solved by linear thresholding operations with the complexity of $\mathcal{O}(m^2)$. Step 6 solves a set of QP problems, leading to the complexity of $\mathcal{O}(mk^{2.5})$.

IV. Experiments

A. Experimental Settings

To demonstrate the effectiveness of the proposed model, we first generated a similarity graph by the proposed approach, and then replaced the graph of an existing graph-based PLL method, to see whether the performance of that graph-based PLL method has been improved. Three graph-based PLL methods were evaluated, and each was configured with the suggested parameters in the literature, i.e.,

TABLE I
CHARACTERISTICS OF THE UCI DATA SETS.

Data Set	Examples	Features	Class Labels
glass	214	9	6
ecoli	336	7	8
movement	360	90	15
vehicle	846	18	4

- PL-KNN [11]: a k-nearest neighbor approach that makes predictions by weighted voting on neighboring instances. [suggested configuration: k=10];
- IPAL [16]: an identification-based method that disambiguates candidate labels by label propagation. [suggested configuration: k=10, α=0.95, T=100];
- LEAF [12]: an average-based PLL method via feature-aware disambiguation. [suggested configuration: k=10, C_1=10, C_2=1].

For our method, λ_1 and λ_2 took values from [0.0001,0.001,...,100] and [0.0001,0.001,...,1], the threshold took values from [0.3,0.4,...,1], and k=10. We evaluated the above methods on both UCI data sets and real-world partial label data sets. Moreover, ten-fold cross-validation was performed on each data set, and the average classification accuracy and the standard deviation were recorded.

B. Controlled UCI Data Sets

Table I summarizes the characteristics of four UCI data sets, i.e., glass, ecoli, movement and vehicle. Following the widely-used partial label data generation protocol [7], [19], these four data sets were used to generate artificial partial label data sets. Specifically, three parameters p, r, ϵ control the generation process, i.e., p controls the proportion of partial label examples, r controls the number of false positive labels, and ϵ controls the probability of a specific positive label occurs with the ground-truth label. Table II shows the classification accuracy of the PLL methods on those four data sets with r=1, p=1, where ORIGIN means a PLL method with the original graph construction method, while PL-TGC indicates that PLL method armed with our graph construction method. Apparently, we can observe from Table II that:

- The classification accuracies have been largely improved by incorporating the proposed graph construction method to different graph-based PLL methods on different data sets. For example, on vehicle data set, classification accuracy increases from 64.7% to 81.2% w.r.t. PL-KNN.
- As our method can promote all the evaluated graph-based PLL algorithms on all the data sets, which suggests it is a general approach for PLL-based graph construction.
- The improvements on PL-KNN are large, while that on IPAL and LEAF are relatively small. As PL-KNN is the most simple graph-based PLL method, the improvements on PL-KNN can better reveal the ability of graph enhancement brought by our method.

TABLE II
CLASSIFICATION ACCURACY OF EACH ALGORITHM ON THE CONTROLLED UCI DATA SETS (r=1).

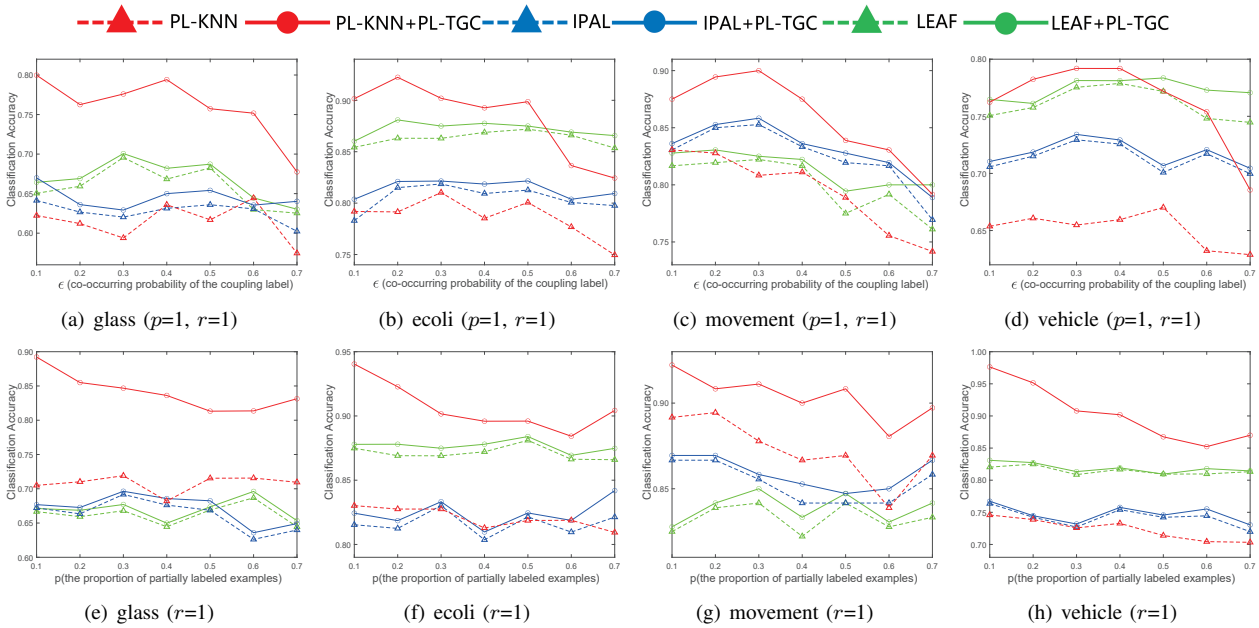
Method		glass	ecoli	movement	vehicle
PL-KNN	ORIGIN	0.647±0.103	0.798±0.064	0.814±0.051	0.647±0.037
	PL-TGC	**0.776±0.068**	**0.911±0.049**	**0.897±0.071**	**0.812±0.019**
IPAL	ORIGIN	0.613±0.078	0.785±0.071	0.872±0.067	0.716±0.034
	PL-TGC	**0.641±0.075**	**0.809±0.081**	**0.881±0.078**	**0.725±0.036**
LEAF	ORIGIN	0.637±0.114	0.872±0.049	0.819±0.062	0.760±0.049
	PL-TGC	**0.656±0.116**	**0.881±0.051**	**0.831±0.061**	**0.772±0.052**

TABLE III
CHARACTERISTICS OF REAL-WORLD PARTIAL LABEL DATA SETS, WHERE AVG. CLS MEANS THE AVERAGE SIZE OF THE CANDIDATE LABEL SET.

Data Set	Examples	Features	Class Labels	Avg. CLs	Task Domain
FG-NET	1002	262	78	7.48	facial age estimation
Lost	1122	108	16	2.23	automatic face naming
MSRCv2	1758	48	23	3.16	object classification
Mirflickr	2780	1536	14	2.76	web image classification
BirdSong	4998	38	13	2.18	bird song classification

To further evaluate the proposed graph construction method, we varied the values of p, r and ϵ to check the performance of the proposed method. Specifically, Figs. 1 (a) - (d) illustrate the classification accuracy of each algorithm as the co-occurring probability ϵ varies from 0.1 to 0.7 with step-size 0.1 (p=1, r=1). Figs. 1 (e) - (h) show the classification accuracy of each algorithm as the proportion p varies from 0.1 to 0.7 with step-size 0.1 when r=1.

- As shown in Figs. 1 (a) - (d), as ϵ increases, the classification accuracies of all the methods decrease, but the improvements brought by our PL-TGC are still significant.
- As shown in Figs. 1 (e) - (h), with the increase of p, the classification accuracies of different methods tend to decrease. But at the same time, the PLL methods incorporating our PL-TGC always outperform the original algorithms. In particular, when r=1 and p=0.4 on the glass data set, the classification accuracy of the original PL-KNN decreases rapidly, but when equipped with our PL-TGC, it can still maintain a high accuracy.
- Among all the 168 cases (14 configurations $\times$ 4 UCI data sets $\times$ 3 methods), PL-KNN+PL-TGC performs better than IPAL+PL-TGC and LEAF+PL-TGC in 98.2% and 87.5%. As PL-KNN is very simple that only uses the neighboring samples captured by the graph to make prediction, we believe such high accuracies are credited to the proposed graph construction method.



(a) glass (p=1, r=1) (b) ecoli (p=1, r=1) (c) movement (p=1, r=1) (d) vehicle (p=1, r=1)

(e) glass (r=1) (f) ecoli (r=1) (g) movement (r=1) (h) vehicle (r=1)

Fig. 1. (a) - (d) illustrate the classification accuracy of each algorithm as ϵ (co-occurring probability of one extra candidate label) increases from 0.1 to 0.7 with p=1, r=1. (e) - (h) illustrate the classification accuracy of each algorithm changes as p (proportion of partially labeled examples) increases with r=1.

TABLE IV
CLASSIFICATION ACCURACY OF EACH ALGORITHM ON THE REAL-WORLD DATA SETS.

Method		FG-NET	FG-NET(MAE3)	FG-NET(MAE5)	Lost	MSRCv2	Mirflickr	BirdSong
PL-KNN	ORIGIN	0.041±0.017	0.319±0.039	0.488±0.039	0.510±0.044	0.395±0.022	0.483±0.026	0.642±0.021
	PL-TGC	**0.065±0.031**	**0.395±0.051**	**0.545±0.051**	**0.561±0.042**	**0.453±0.037**	**0.498±0.024**	**0.707±0.017**
IPAL	ORIGIN	0.059±0.023	0.333±0.034	0.492±0.039	0.682±0.046	0.532±0.016	0.515±0.043	0.732±0.020
	PL-TGC	**0.072±0.018**	**0.352±0.029**	**0.537±0.051**	**0.707±0.035**	**0.542±0.016**	**0.527±0.038**	**0.742±0.021**
LEAF	ORIGIN	0.077±0.027	0.460±0.052	0.611±0.056	0.744±0.042	0.533±0.030	0.646±0.030	0.736±0.021
	PL-TGC	**0.081±0.025**	**0.464±0.065**	**0.615±0.047**	**0.765±0.052**	**0.535±0.024**	**0.649±0.028**	**0.740±0.020**

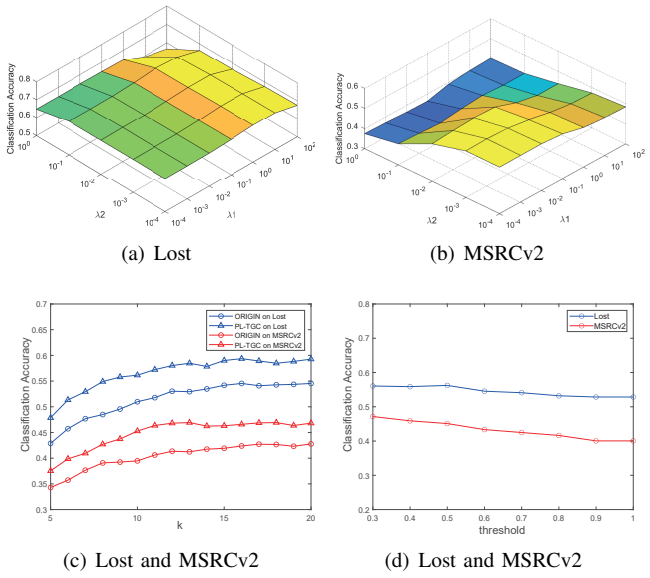
C. Real-World Data Sets

Table III summarizes the characteristics of real-world partial label data sets, which are collected from various tasks and domains including FG-NET [20] for facial age estimation, Lost [7] for automatic face naming from images or videos, MSRCv2 [10] for object classification, Mirflickr [21] for web image classification and BirdSong [22] for bird song classification. As the average size of the candidate label set (Avg. CLs) of FG-NET is large, which could cause low classification accuracy on the test set. To better evaluate this facial age estimation task, we employed the mean absolute error (MAE) to calculate two extra evaluation indicators FG-NET (MAE3) and FG-NET (MAE5), i.e., the test examples are considered to be correctly classified if the difference between the predicted age and the ground-truth age is no more than 3/5 years.

Table IV demonstrates the classification accuracies of different methods on real-world data sets. Similar to results on the controlled UCI data sets, by incorporating PL-TGC the classification accuracies of different graph-based PLL methods on real-world data sets have all been improved. Especially, by incorporating PL-TGC, the classification accuracy of PL-KNN on five real-world data sets is improved by 4.9% in average. Those observations demonstrate the proposed graph construction method is also fit for real-world PLL tasks.

D. Further Analysis

PL-TGC has four parameters, i.e., λ_1, λ_2, threshold value and k. The performances of PL-TGC under different parameter configurations in Lost and MSRCv2 are shown in Fig. 2. As shown in Figs. 2 (a) - (b), our PL-TGC has a smooth region near the optimal value, which indicates the stability of our method to λ_1 and λ_2. Fig. 2 (c) indicates as k increases, the performance of PL-KNN generally gets better. When k=10, the performance of PL-KNN is relatively stable, so k was fixed to 10 in the experiments. Fig. 2 (d) shows that our algorithm is relatively robust to the threshold.



(a) Lost (b) MSRCv2

(c) Lost and MSRCv2 (d) Lost and MSRCv2

Fig. 2. Parameter sensitivity analysis. (a) Classification accuracies of IPAL+PL-TGC on Lost by varying λ_1 and λ_2; (b) Classification accuracies of PL-TGC on MSRCv2 by varying λ_1 and λ_2; (c) Classification accuracies of PL-KNN+PL-TGC on Lost, MSRCv2 by varying k; (d) Classification accuracies of PL-KNN+PL-TGC on Lost, MSRCv2 by varying the threshold.

TABLE V
ABLATION STUDY OF OUR METHOD WITH PL-KNN

Data Set	ORIGIN	PL-TGC-1	PL-TGC-2	PL-TGC
glass	0.571±0.115	0.575±0.111	0.682±0.092	**0.691±0.088**
ecoli	0.723±0.053	0.749±0.061	0.797±0.053	**0.812±0.048**
movement	0.739±0.063	0.758±0.056	0.811±0.058	**0.817±0.049**
vehicle	0.533±0.051	0.560±0.054	0.629±0.064	**0.632±0.058**

PL-TGC contains three components, i.e., local self-representation, adversarial term regarding λ_1, and the dissimilarity propagation term regarding λ_2. Here, we conduct an ablation study on the controlled UCI data sets with r=2 to check the effectiveness of the involved terms. Specifically, ORIGIN means the performance of PL-KNN with the original graph construction method. PL-TGC-1 (resp. PL-TGC-2) indicates the proposed model with λ_1=0 (resp. λ_2=0) and uses PL-KNN as the classification model. PL-TGC denotes the full version of our model with PL-KNN as the classifier. Table V shows the corresponding results, where we can find that both the involved adversarial term and the dissimilarity propagation term are helpful in improving classification accuracy and taking both of them into account is the best choice.

V. CONCLUSION

In this paper, the problem of PLL tailored graph construction was investigated. The proposed model first extracted a few limited but high reliable dissimilarity relationships from the label space, and enhanced it by propagation. Then the enhanced dissimilarity matrix was used to refine the similarity matrix extracted from the feature space by an adversarial prior. The learned graph was specifically designed for PLL and extensive experiments on artificial as well as real-world partial label data sets validated that it can promote various graph-based PLL methods.

VI. ACKNOWLEDGMENTS

This work was supported by the National Natural Science Foundation of China under Grant U24A20322.

REFERENCES

[1] C. Tang and M. Zhang, "Confidence-rated discriminative partial label learning," in *Proceedings of the Thirty-First AAAI Conference, 2017*, S. Singh and S. Markovitch, Eds., 2017, pp. 2611–2617.
[2] L. Feng and B. An, "Partial label learning by semantic difference maximization," in *Proceedings of the Twenty-Eighth IJCAI*, 2019, pp. 2294–2300.
[3] Y. Jia, F. Yang, and Y. Dong, "Partial label learning with dissimilarity propagation guided candidate label shrinkage," in *Advances in Neural Information Processing Systems 36*, 2023.
[4] F. Yang, J. Cheng, H. Liu, Y. Dong, Y. Jia, and J. Hou, "Mixed blessing: Class-wise embedding guided instance-dependent partial label learning," *CoRR*, vol. abs/2412.05029, 2024.
[5] Y. Jia, X. Peng, R. Wang, and M. Zhang, "Long-tailed partial label learning by head classifier and tail classifier cooperation," in *Thirty-Eighth AAAI Conference*, 2024, pp. 12 857–12 865.
[6] J. Jiang, Y. Jia, H. Liu, and J. Hou, "Fairmatch: Promoting partial label learning by unlabeled samples," in *ACM SIGKDD International Conference*, 2024, pp. 1269–1278.
[7] T. Cour, B. Sapp, and B. Taskar, "Learning from partial labels," *Journal of Machine Learning Research*, vol. 12, pp. 1501–1536, 2011.
[8] Z. Zeng, S. Xiao, K. Jia, T. Chan, S. Gao, D. Xu, and Y. Ma, "Learning by associating ambiguously labeled images," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 708–715.
[9] J. Luo and F. Orabona, "Learning from candidate labeling sets," in *Advances in Neural Information Processing Systems 23: 24th Annual Conference*, 2010, pp. 1504–1512.
[10] L. Liu and T. G. Dietterich, "A conditional multinomial mixture model for superset label learning," in *Advances in Neural Information Processing Systems 25: 26th Annual Conference*, 2012, pp. 557–565.
[11] E. Hüllermeier and J. Beringer, "Learning from ambiguously labeled examples," *Intelligent Data Analysis*, vol. 10, no. 5, pp. 419–439, 2006.
[12] M. Zhang, B. Zhou, and X. Liu, "Partial label learning via feature-aware disambiguation," in *Proceedings of the 22nd ACM SIGKDD International Conference*, 2016, pp. 1335–1344.
[13] R. Jin and Z. Ghahramani, "Learning with multiple labels," in *Advances in Neural Information Processing Systems 15*, 2002, pp. 897–904.
[14] N. Nguyen and R. Caruana, "Classification with partial labels," in *Proceedings of the 14th ACM SIGKDD International Conference*, 2008, pp. 551–559.
[15] F. Yu and M. Zhang, "Maximum margin partial label learning," *Machine Learning*, vol. 106, no. 4, pp. 573–593, 2017.
[16] M. Zhang and F. Yu, "Solving the partial label learning problem: An instance-based approach," in *Proceedings of the Twenty-Fourth IJCAI*, 2015, pp. 4048–4054.
[17] L. Feng and B. An, "Leveraging latent label distributions for partial label learning," in *Proceedings of the Twenty-Seventh IJCAI*, 2018, pp. 2107–2113.
[18] D. Wang, L. Li, and M. Zhang, "Adaptive graph guided disambiguation for partial label learning," in *Proceedings of the 25th ACM SIGKDD International Conference, KDD 2019*, 2019, pp. 83–91.
[19] Y. Chen, V. M. Patel, R. Chellappa, and P. J. Phillips, "Ambiguously labeled learning using dictionaries," *IEEE Transactions on Information Forensics and Security*, vol. 9, no. 12, pp. 2076–2088, 2014.
[20] G. Panis and A. Lanitis, "An overview of research activities in facial age estimation using the FG-NET aging database," in *European Conference on Computer Vision*, vol. 8926, 2014, pp. 737–750.
[21] M. J. Huiskes and M. S. Lew, "The MIR flickr retrieval evaluation," in *Multimedia Information Retrieval*, 2008, pp. 39–43.
[22] F. Briggs, X. Z. Fern, and R. Raich, "Rank-loss support instance machines for MIML instance annotation," in *The 18th ACM SIGKDD International Conference*, 2012, pp. 534–542.