Dimensionality Reduction for Partial Label Learning: A Unified and Adaptive Approach

Xiang-Ru Yu, Deng-Bao Wang, and Min-Ling Zhang, *Senior Member, IEEE*

Abstract—Partial label learning learns from instances with weak supervision, where each instance is associated with a set of candidate labels, among which only one is valid. Recently, dimensionality reduction has emerged as an effective preprocessing strategy to improve generalization performance. Existing approaches mainly tackle this problem through supervised or unsupervised dimensionality reduction. However, the former requires ground-truth labels, which are concealed in candidate label sets. Consequently, methods in this line may suffer from overfitting due to false positive labels in candidate label set. Conversely, the latter overlooks weakly supervised information in training instances, leading to performance degradation. In this paper, we propose an approach called *partial label Dimensionality Reduction via Adaptive Weight (*DRAW*)* to leverage the strengths of Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA). Specifically, our approach tends to exploit unsupervised and data-driven nature of PCA to capture underlying structure of instances in initial stage. As the ground-truth label is gradually identified, our method increasingly relies on the discriminative ability of LDA to enhance the separation between different classes. Through extensive experiments on diverse partial label datasets, we validate that the proposed dimensionality reduction approach significantly improves classification performance of well-established partial label learning algorithms.

Index Terms—Partial label learning, dimensionality reduction, linear discriminant analysis, principal component analysis.

1 INTRODUCTION

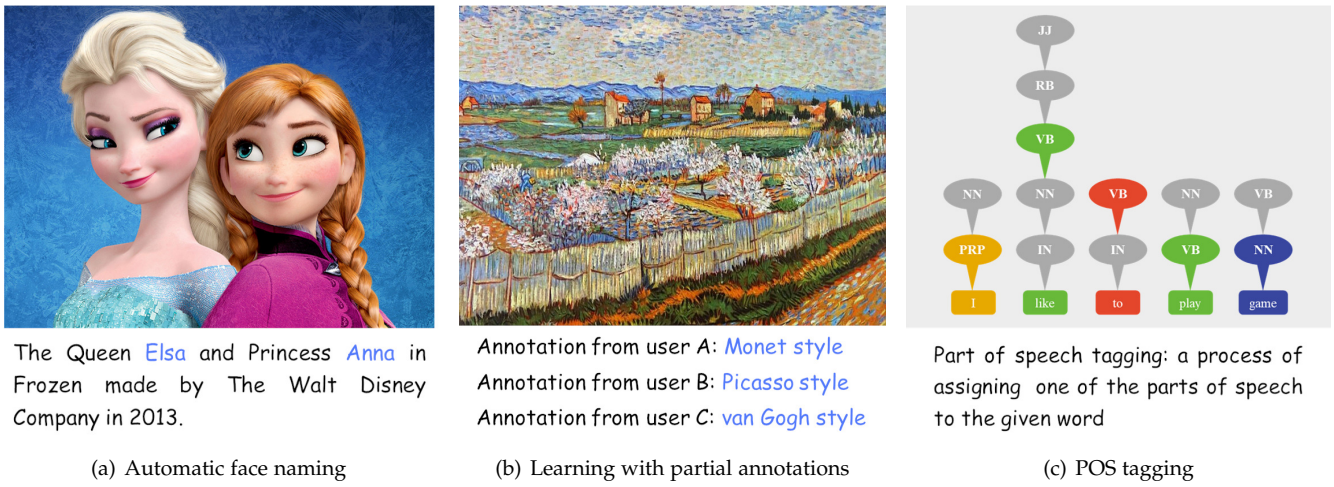
MULTI-CLASS classification has achieved great success in many real-world tasks such as image classification and natural language processing [1], [2], [3], where each object is represented as a single instance with a explicit label. However, constrained by various factors, only limited supervision information can be obtained from training data in many real-world applications, where the supervision is usually incomplete, inexact(only coarse-fined label) or inaccurate(label noise). Weakly supervised learning focuses on dealing with these data with limited supervision [4]. Partial label learning is one of the popular weakly supervised paradigms [5], [6], where each instance is associated with a set of candidate labels among which only one is ground-truth label [7], [8]. Partial label learning aims to induce a multi-class classification model from training data with ambiguous supervision. In recent years, the need to learn from partial labeled data naturally arises in many real-world applications. For example, as shown in Fig. 1(a), in automatic face naming, a news document treats each face detected in the picture as an instance while those names extracted from the corresponding caption as the candidate label set, but the actual correspondence between the instance and candidate labels is not unknown [9], [10]; for the crowd-sourcing data in Fig. 1(b), due to the difference in professional ability of annotators, the annotation results commonly constitute a candidate label set instead the only valid label, while the ground-truth label resides in those labels [11]. By now, the partial label learning has been successfully applied to face age estimation [12], multimedia content analysis [13], [14], ecoinformatics [15], [16], part-of-speech tagging [17] as shown in Fig. 1(c), etc.

Formally speaking, let $\mathcal{X} = \mathbb{R}^d$ denote the d-dimensional instance space and $\mathcal{Y} = \{l_1, l_2, \ldots, l_q\}$ denote the label space with q class labels. Given the partial label training set $\mathcal{D} = \{(\boldsymbol{x}_i, S_i) \mid 1 \leq i \leq m\}$ where $\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional feature vector $(x_{i1}, x_{i2}, \ldots, x_{id})^\mathsf{T}$ and $S_i \subseteq \mathcal{Y}$ is the candidate label set associated with $\boldsymbol{x}_i$. The task of partial label learning is to induce a *multi-class* classifier $f : \mathcal{X} \mapsto \mathcal{Y}$ from the training set $\mathcal{D}$. In partial label learning, the basic assumption lies in that the ground-truth label y_i for $\boldsymbol{x}_i$ resides in its candidate label set S_i (i.e. $y_i \in S_i$) which is inaccessible during training phase.

It is widely believed that the overfitting problem is ubiquitous in machine learning, especially when dealing with training data that contains noise. For these learning systems, dimensionality reduction provides an effective view to improve the generalization ability. In partial label learning, recent work leverages the feature-induced label disambiguation to recover the ground-truth labels. However, models tend to fit false positive labels associated with samples in candidate label sets, making it challenging to generalize to the test data. For these partial label learning algorithms, high-dimensional data often suffers from the curse of dimensionality, where the distribution of training instances becomes sparse. At the same time, high-dimensional data may contain noisy and redundant features, which may harm feature-induced label disambiguation. Therefore, we aim to

- *Xiang-Ru Yu is with the School of Cyber Science and Engineering, Southeast University, Nanjing 210096, China, and the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China. E-mail: yuxr@seu.edu.cn.*
- *Deng-Bao Wang, and Min-Ling Zhang are with the School of Computer Science and Engineering, Southeast University, Nanjing 210096, China, and the Key Laboratory of Computer Network and Information Integration (Southeast University), Ministry of Education, China. E-mail: wangdb @seu.edu.cn,zhangml@seu.edu.cn. (Corresponding author: Min-Ling Zhang)*

Manuscript received May 22, 2023;



(a) Automatic face naming (b) Learning with partial annotations (c) POS tagging

Fig. 1. Applications of partial label learning in real world. (a) In automatic face naming, names extracted from the caption serve as candidate labels for each face detected in the image or frame in video. (b) For crowd-sourcing data, the labeling result given by annotators constitutes candidate label set, among which only one label is valid and unknown. (c) In part-of-speech (POS) tagging, the target word with contextual features can be defined as instance, while the possible part of speech about target word form the candidate label set.

explore dimensionality reduction to solve this problem. Because dimensionality reduction not only attains a more compact representation, but also increases the sample density. Accordingly, it is helpful to make the label disambiguation process more accurate, and hence enhances the performance of partial label learning algorithms. Although it is desirable to explore dimensionality reduction in partial label learning to improve the generalization performance, the ambiguous supervision of partial labels limits the application of dimensionality reduction algorithms, which makes partial label dimensionality reduction rarely investigated. To the best of our knowledge, DELIN [18] and CENDA [19] are the only partial label dimensionality reduction algorithms. DELIN adapts LDA technique to maximize the between-class covariance matrix while minimize the within-class covariance matrix. CENDA employs the Hilbert-Schmidt Independence Criterion (HSIC) to maximize the dependence between the feature space and label information. However, in essence, DELIN and CENDA directly treat partial label dimensionality reduction as supervised learning, thus, the algorithm may suffer from overfitting due to the false positive labels in candidate label set.

To solve the potential drawbacks of DELIN and CENDA, we attempt to decrease the impact of overfitting caused by the ambiguous supervision in partial label learning via adaptive combination of unsupervised and supervised dimensionality reduction. On the one hand, supervised dimensionality reduction needs explicit supervision, however, partial label learning usually suffers from overfitting due to the the ambiguous supervision. Unsupervised dimensionality reduction is not affected by the quality of supervision information for its unsupervised nature, thus, unsupervised dimensionality reduction is helpful to reduce the impact of overfitting. On the other hand, unsupervised dimensionality reduction induces projection matrix by only global structure in feature space without supervision information, accordingly, its effectiveness is susceptible to the changement of scales, supervised dimensionality reduction is helpful to improve the stability of unsupervised one due to the supervised nature. To better exploit both of these two kind of dimensionality reduction methods, we propose a unified partial label dimensionality reduction approach named DRAW, i.e., *partial label Dimensionality Reduction via Adaptive Weighting*. DRAW combines the supervised and unsupervised dimensionality reduction in adaptive manner to dynamically balance the contribution of this two strategies.

Specifically, linear discriminant analysis (LDA) [20] is a popular algorithm for supervised dimensionality reduction, while principal component analysis (PCA) [21] is a typical algorithm of unsupervised dimensionality reduction. DRAW combines the above two methods in a unified framework to identify the projection matrix with adaptive weight, which can diminish the impact of noisy labels on identifying the projection matrix when ground-truth label is not clear. With the confirmation of ground-truth label, the weight of supervised dimensionality reduction gradually increases. In each iteration, DRAW alternates between dimensionality reduction and candidate label disambiguation. Comprehensive experiments conducted on real-world and synthetic partial label data sets validate the effectiveness of DRAW to improve the generalization performance of the state-of-the-art partial label learning algorithms.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work on partial label learning and dimensionality reduction. Section 3 presents the technical procedure of proposed approach DRAW. Section 4 reports the detailed experimental setting and comparative results of experiments. Finally, Section 5 concludes and indicates future work.

2 RELATED WORK

Partial label learning [22], [23] learns from training examples with ambiguous supervision, where the ground-truth label is concealed in the candidate label set, while it is inaccessible during training phase. Therefore, partial label learning can be regarded as a emerging weakly supervised learning paradigm, other well-established weakly supervised learning frameworks include *multi-label learning*

[24], *semi-supervised learning* [25], *multi-instance learning* [26], *active learning* [27] and *multi-instance multi-label learning* [28].

To clarify them clearly, we lay out the corresponding learning scenarios. Multi-label learning learns from training data where each instance is assigned to multiple valid labels, while for partial label learning, only one is valid. In semi-supervised learning, the instance is unlabeled or assigned to explicit labels, while in partial label learning, the supervision is ambiguous. In multi-instance learning, the training data is composed of multiple bags where each one contains multiple instances, and the label is assigned at the level of bag, while in partial label learning, the label is assigned to instance. Active learning allows for human intervention, which assumes that there is an *'oracle'*, such as a human expert, the ground-truth label of unlabeled examples can be queried from the oracle, while in partial label learning, human intervention is not involved. Multi-instance multi-label learning [29], [30] is a generalized framework of multi-instance learning and multi-label learning.

Partial Label Learning. The main difficulty for partial label learning resides in that the only ground-truth label is concealed in the candidate label set. Accordingly, disambiguation [31], [32] serves as an intuitive approach, which aims to identify the ground-truth label from the candidate label set. In general, disambiguation is commonly achieved by two main strategies, namely, averaging-based disambiguation [33], [34] and identification-based disambiguation [35], [36], [37]. Following averaging-based disambiguation, each label in candidate label set is treated equally, and the prediction is made by averaging the model outputs on each candidate label. For parametric model $g(x)$, the averaged output over candidate label set $\frac{1}{|S_i|}\sum_{y\in S_i} g_y(\boldsymbol{x}_i)$ is distinguished from the outputs from non-candidate labels [38], i.e. $g_y(\boldsymbol{x}_i)$, where $y \notin S_i$. Despite the averaging-based disambiguation is intuitive, the output of ground-truth label is prone to be overwhelmed by the false positive labels in candidate label set. Consequently, another strategy is to identify the ground-truth label. Following this strategy, identification-based disambiguation treats the ground-truth label as a latent variable, and the Expectation Maximization (EM) procedure is applied to refine the estimation of latent variable via optimizing the objective function in iterations. Generally, the objective function is defined based on the maximum likelihood criterion $log(\sum_{y\in S_i} p(y|\boldsymbol{x_i}, \boldsymbol{\theta}))$ or maximum margin criterion $max_{y_j\in S_i} f(\boldsymbol{x}_i, y_j) - max_{y_k\notin S_i} f(\boldsymbol{x}_i, y_k)$. However, for identification-based disambiguation, the recovering label might turn out to be a false positive label rather than the ground-truth one.

Contrary to disambiguation, disambiguation-free methods adapt popular learning techniques to solve partial label problem directly. Generally, the partial label data set $\mathcal{D}$ is converted into binary classification data sets by exploring the explicit opposite relationship between candidate labels and non-candidate labels $\mathcal{Y} \setminus S_i$. After, multi-class classifier $f : \mathcal{X} \mapsto \mathcal{Y}$ is directly induced from the converted data set. Error-correcting outputs code (ECOC) [39] treats the candidate label set S_i as a entirety, and the partial label instance is regarded as positive or negative if S_i entirely falls into the coding dichotomy of ECOC coding matrix. Similarly, binary decomposition mechanism [40] takes the label pair (y_j, y_k) to induce classifier, where the following constraint must be satisfied: $y_j \in S_i$ and $y_k \in \mathcal{Y} \setminus S_i$.

Existing works mainly focus on the manipulation on label space. As an effective method to improve the generalization ability of learning algorithms, dimensionality reduction exploits the manipulation on feature space [41].

Dimensionality Reduction. Depending on whether the label information is used, dimensionality reduction can be classified into two categories, namely, unsupervised and supervised. For unsupervised dimensionality reduction [42], PCA is a representative algorithm, which identifies the projection matrix by maximizing the variance of projected data. There are some classic methods for manifold learning that can be used for dimensionality reduction, including isometric mapping (ISOMAP) [43], locally linear embedding (LLE) [44], Laplacian Eigenmaps (LE) [45] and locality preserving projection (LPP) [46]. Metric embedding has also been extensively studied in the past few decades. For example, Lipschitz embedding [47], [48] maps data points into a lower-dimensional metric space by leveraging a set of pivot points to approximate the distance in source metric space. Bourgain theorem further proves that every metric space can be embedded into a nice normed space with only a logarithmic loss in distortion. Tree embedding aims to embed the points into a spanning tree, and Hierarchically Separated Tree (HST) is a popular data structure to embed the original metric space into a tree-based metric space [49], [50], [51]. Among them, dynamic programming-based method is designed to reduce the time and space complexity in construction, and Hierarchically Separated Forest (HSF) is further proposed to adapt the insertion of new points [49]. Additionally, DCsam strives to reduce the time complexity and ensure the distortion bound [50]. For partial label learning, the supervision from partial label instances is more rich in semantic against unsupervised instances. Accordingly, unsupervised dimensionality reduction algorithms completely ignores the supervision from candidate labels, which may result in degraded classification performance.

For supervised dimensionality reduction, LDA is a representative algorithm, which identifies the projection matrix by maximizing the intra-class similarity and minimizing the inter-class similarity simultaneously. Some advances on supervised dimensionality reduction have been studied in the past few decades, such as canonical correlation analysis (CCA), partial least square (PLS), latent semantic indexing [52], etc. In multi-class learning and multi-label learning, dimensionality reduction has been widely investigated, where the instance is assigned to one or multiple valid labels. Therefore, due to the limitation of ambiguous supervision, for partial label learning, dimensionality reduction is rarely investigated. To the best of our knowledge, DELIN [18] and CENDA [19] are the only dimensionality reduction algorithms that adapt partial label learning framework. Aiming at the candidate label set, DELIN adapts LDA technique to induce the projection matrix via confidence label vector instead of explicit label. Similarly, CENDA employs the Hilbert-Schmidt Independence Criterion (HSIC) to derive projection matrix via confidence-based label.

Regardless of DELIN and CENDA, they attempt to adapt supervised dimensionality reduction to partial label learning, which ignores the impact of false positive labels in candidate label set. Therefore, DRAW identifies the projec-

tion matrix via combining the supervised and unsupervised dimensionality reduction via adaptive weight. Specifically, in the initial stage, the supervised dimensionality reduction is easy to be misled by the false positive labels, and the unsupervised dimensionality reduction is helpful to mitigate it. In the iterative process, the ground-truth label is gradually identified, and the reliability of supervision information increases step by step. Accordingly, we adjust the weight to balance the contribution of unsupervised dimensionality reduction and supervised one adaptively. Finally, the supervised dimensionality reduction dominates the induction of projection matrix.

3 The Proposed Approach

Dimensionality reduction focuses on finding a projection matrix $\mathbf{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_{d'}] \in \mathbb{R}^{d \times d'}$ ($d' \ll d$), which maps the training data $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m] \in \mathbb{R}^{d \times m}$ with d-dimension features into the projected feature space with d'-dimension features, $\mathbf{X}' = \mathbf{P}^\top \mathbf{X}$. In the machine learning community, both supervised and unsupervised dimensionality reduction setting have been extensively studied. In the context of partial label learning, we attempt to combine strategies from both of these two settings to address the ambiguous supervision. Specifically, we consider two representative methods, namely Principal Component Analysis (PCA) for unsupervised dimensionality reduction and Linear Discriminant Analysis (LDA) from supervised dimensionality reduction. Before introducing our approach, we provide a brief overview of these two methods.

3.1 Preliminaries

Principal Component Analysis (PCA). As a widely-used unsupervised dimensionality reduction method [53], [54], PCA aims to find the maximum-variance direction of training data, and the optimization objective of projection matrix can be expressed in the following form:

$$\mathbf{P}^{PCA} = \underset{\mathbf{P} \in \mathbb{R}^{d \times d'}}{\arg\max} \left[\frac{tr(\mathbf{P}^T \mathbf{S}_t \mathbf{P})}{tr(\mathbf{P}^T \mathbf{I}_d \mathbf{P})} \right] \tag{1}$$

where $\mathbf{I}_d \in \mathbb{R}^{d \times d}$ is an identity square matrix, $\mathbf{P}$ denotes the projection matrix, and $\mathbf{S}_t$ is the total scatter matrix, which is shown as below:

$$\mathbf{S}_t = \sum_{i=1}^{m} (\boldsymbol{x}_i - \boldsymbol{\mu})(\boldsymbol{x}_i - \boldsymbol{\mu})^\top \tag{2}$$

where $\boldsymbol{x}_i$ is the i-th instance vector with d dimensional features, $\boldsymbol{\mu}$ is the global mean vector of all data points, which is specified as:

$$\boldsymbol{\mu} = \frac{\sum_{i=1}^{m} \boldsymbol{x}_i}{m} \tag{3}$$

According to the above definition, it is easy to find that PCA only refers to data points without label information. Therefore, without adjustment, it can naturally be used to induce the projection matrix from partial label data, and map the data from original feature space into the projected feature space. Nonetheless, PCA completely ignores the weakly-supervised information from candidate label set of partial label data, which may result in degraded classification performance.

Linear Discriminant Analysis (LDA). Compared with PCA, LDA is a popular supervised dimensionality reduction technique. Given the multi-class data set $\mathcal{D} = \{(\boldsymbol{x}_i, Y_i) \mid 1 \leq i \leq m\}$, where $\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional feature vector $(x_{i1}, x_{i2}, \ldots, x_{id})^\top$ and $Y_i \in \mathcal{Y}$ is the corresponding ground-truth label of $\boldsymbol{x}_i$.

LDA aims to maximize the generalized Rayleigh quotient between the between-class scatter matrix and within-class scatter matrix, i.e. maximizing the between-class covariance while minimizing the within-class covariance.

Let $\mathbf{S}_b$ and $\mathbf{S}_w$ represent the between-class scatter matrix and within-class scatter matrix respectively as follows:

$$\mathbf{S}_b = \sum_{j=1}^{q} n_j \cdot (\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^\top \tag{4}$$

where n_j is the number of labeled instances with label j, $\boldsymbol{\mu}$ denote the global mean vector, and the $\boldsymbol{\mu}_j$ is the mean vector from the instances with l_j.

$$\mathbf{S}_w = \sum_{j=1}^{q} \sum_{i=1}^{m} (\boldsymbol{x}_i - \boldsymbol{\mu}_j)(\boldsymbol{x}_i - \boldsymbol{\mu}_j)^\top \tag{5}$$

Obviously, the total scatter matrix in Eq.(2) is equal to the sum of between-class scatter matrix and with-class scatter matrix,

$$\mathbf{S}_t = \mathbf{S}_w + \mathbf{S}_b \tag{6}$$

Note that the total scatter matrix is consistent with PCA.

Based on the above scatter matrix, the optimization objective of LDA is defined as:

$$\mathbf{P}^{LDA} = \underset{\mathbf{P} \in \mathbb{R}^{d \times d'}}{\arg\max} \left[\frac{tr(\mathbf{P}^T \mathbf{S}_b \mathbf{P})}{tr(\mathbf{P}^T \mathbf{S}_w \mathbf{P})} \right] \tag{7}$$

According to the above definitions, LDA requires the ground-truth label to determine the between-class and within-class scatter matrix. However, the ground-truth label is concealed in the candidate label set for partial label learning, which is unknown in training stage. Therefore, LDA may suffer from overfitting due to the false positive labels in candidate label set.

3.2 A Unified Dimensionality Reduction Framework

As shown in Eq. (1) and (7), despite of the different formulations of PCA and LDA, their objective functions are highly similar to each other, which makes it possible to combine them. In particular, the optimization objective functions of PCA and LDA can be formulated in a unified framework:

$$\mathbf{P}^{opt} = \underset{\mathbf{P} \in \mathbb{R}^{d \times d'}}{\arg\max} \left[\frac{tr(\mathbf{P}^T \mathbf{B} \mathbf{P})}{tr(\mathbf{P}^T \mathbf{C} \mathbf{P})} \right] \tag{8}$$

where $\mathbf{B}$ and $\mathbf{C}$ are two matrices which are identified according to the definition of specific dimensionality reduction algorithms. Specifically, we can define the between-class and within-class scatter matrix as shown in the following equations:

$$\mathbf{B} = (1 - \alpha)\mathbf{S}_b + \alpha \mathbf{S}_t \tag{9}$$

$$\mathbf{C} = (1 - \alpha)\mathbf{S}_w + \alpha \mathbf{I}_d \tag{10}$$

TABLE 1
The pseudo-code of DRAW.

Inputs:

$\mathcal{D}$: the partial label training data set $\{(\boldsymbol{x}_i, S_i) \mid 1 \le i \le m\}$ ($\mathcal{X} \in \mathbb{R}^d, \mathcal{Y} = \{l_1, l_2, \ldots, l_q\}, \boldsymbol{x}_i \in \mathcal{X}, S_i \subseteq \mathcal{Y}$)

d': the number of dimension after dimensionality reduction

k: the number of nearest neighbors used for candidate label disambiguation

α: the trade-off parameter to balance LDA and PCA

s: the decay parameter to control the speed to decrease the weight of PCA

Outputs:

$\mathbf{P}$: the induced $d \times d'$ projection matrix via the proposed approach DRAW

$\mathcal{D}'$: the transformed lower-dimensional partial label training set $\{(\boldsymbol{x}'_i, S_i) \mid 1 \le i \le m\}$

Process:

Initialize the $m \times q$ label confidence matrix $\mathbf{F}$ according to Eq.(12);
Cascade the training data into the instance matrix $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m]$;
Calculate the global mean vector $\boldsymbol{\mu}$ according to Eq.(3);
repeat
  Calculate the class-wise mean vector $\boldsymbol{\mu}_j$ $(1 \le j \le q)$ according to Eq.(15);
  Induce the total scatter matrix $\mathbf{S}_t$ and between-class scatter matrix $\mathbf{S}_b$ according to Eq.(2) and Eq.(14) respectively;
  Form the projection matrix $\mathbf{P}^{\text{DRAW}} = \arg\max_{\mathbf{P} \in \mathbb{R}^{d \times d'}} \left[\frac{tr(\mathbf{P}^T \mathbf{S}_b \mathbf{P})}{(1-\alpha) \cdot tr(\mathbf{P}^T \mathbf{S}_w \mathbf{P}) + \alpha \cdot tr(\mathbf{P}^T \mathbf{I}_d \mathbf{P})} \right]$;
  Solve the problem of Eq.(11), and derive the projection matrix $\mathbf{P} = [\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_{d'}]$ by concatenating the eigenvectors corresponding to the top d' eigenvalues via Eq.(16);
  Map the partial label training data into lower-dimensional feature space $\mathcal{D}' = \{(\boldsymbol{x}'_i, S_i) \mid \boldsymbol{x}'_i = \mathbf{P}^\top \boldsymbol{x}_i, 1 \le i \le m\}$;
  for i=1 to m **do**
    Identify the k-nearest neighbors of $\boldsymbol{x}'_i$ in $\mathcal{D}'$ as $\mathcal{N}(\boldsymbol{x}'_i)$;
  end for
  Calculate the $m \times m$ weighted matrix $\mathbf{W}$ via kNN aggregation according to the manifold structure in feature space and label space via Eq.(17);
  Derive the updated label confidence matrix $\mathbf{F}'$ according to the weight matrix via Eq.(19) and Eq.(20);
  Assign $\mathbf{F} = \mathbf{F}'$;
  Adjust the value of trade-off parameter α via decay constant as $\alpha = \alpha \cdot s$;
until convergence
Return the learned partial label projection matrix $\mathbf{P}$ and transformed lower-dimensional partial label data set $\mathcal{D}'$.

where $\alpha \in [0, 1]$ is the trade-off parameter to balance the weight between PCA and LDA. When $\alpha = 0$, Eq. (8) is reduced to LDA, while taking $\alpha = 1$ Eq. (8) is equivalent to PCA. Here, we can simplify the formulation by realizing $\mathbf{B}$ as $\mathbf{S}_b$. Accordingly, the objective function of our unified dimensionality reduction framework DRAW can be represented as follows:

$$\mathbf{P}^{\text{DRAW}} = \arg\max_{\mathbf{P} \in \mathbb{R}^{d \times d'}} \left[\frac{tr(\mathbf{P}^T \mathbf{S}_b \mathbf{P})}{(1-\alpha) \cdot tr(\mathbf{P}^T \mathbf{S}_w \mathbf{P}) + \alpha \cdot tr(\mathbf{P}^T \mathbf{I}_d \mathbf{P})} \right] \quad (11)$$

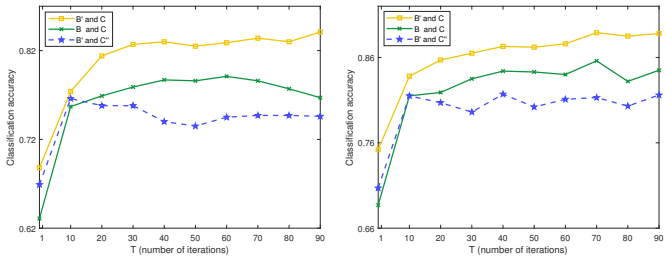
In Fig. 2, we compare the performance of our framework with different forms. It is shown that the simplified form achieves more stable performance than the above form that takes $\mathbf{B}$ by Eq. (9).

Due to the lack of label assignment, in our framework, a generated label confidence matrix $\mathbf{F} = [\boldsymbol{f}_1; \boldsymbol{f}_2; \ldots; \boldsymbol{f}_m] \in \mathbb{R}^{m \times q}$ is used instead to compute $\mathbf{S}_w$ and $\mathbf{S}_b$. The value of element f_{ij} reflects the confidence on that label l_j is the ground-truth label for instance $\boldsymbol{x}_i$, which satisfies the constraint, $\sum_{j=1}^{q} f_{ij} = 1$ for $i \in [1, 2, \ldots, m]$.

Our dimensionality reduction method performs as a iterative procedure, thus the label confidence matrix $\mathbf{F}$ can be dynamically generated during different iterations. In the initial stage, the label confidence vector is initialized with equal confidence for each label in candidate label set as follows,

$$\forall\, 1 \le i \le m,\ 1 \le j \le q: \qquad f_{ij} = \begin{cases} \frac{1}{|S_i|}, & \text{if } l_j \in S_i \\ 0, & \text{otherwise} \end{cases} \quad (12)$$

Accordingly, based on the label confidence matrix $\mathbf{F}$, the within-class scatter matrix and between-class scatter matrix are defined as:



(a) PL-KNN (b) IPAL

Fig. 2. The classification accuracy coupled with different objective functions in dimensionality reduction on Lost, where $\mathbf{B}' = \mathbf{S}_b$, $\mathbf{C}'' = \mathbf{S}_w$, while the corresponding objective function is $\mathbf{P}^{\text{DRAW}}$(yellow), $\mathbf{P}^{opt}$(green) and $\mathbf{P}^{LDA}$(blue). (a) PL-KNN coupled with $\mathbf{P}^{LDA}$, $\mathbf{P}^{opt}$ and $\mathbf{P}^{\text{DRAW}}$ via objective function in Eq.(7), Eq.(8) and simplified objective function in Eq.(11). (b) IPAL coupled with $\mathbf{P}^{LDA}$, $\mathbf{P}^{opt}$ and $\mathbf{P}^{\text{DRAW}}$.

$$\mathbf{S}_w = \sum_{j=1}^{q}\sum_{i=1}^{m} f_{ij}\cdot(\boldsymbol{x}_i-\boldsymbol{\mu}_j)(\boldsymbol{x}_i-\boldsymbol{\mu}_j)^\top \quad (13)$$

$$\mathbf{S}_b = \sum_{j=1}^{q}\left(\sum_{i=1}^{m} f_{ij}\right)\cdot(\boldsymbol{\mu}_j-\boldsymbol{\mu})(\boldsymbol{\mu}_j-\boldsymbol{\mu})^\top \quad (14)$$

where the $\boldsymbol{\mu}_j$ is induced as:

$$\boldsymbol{\mu}_j = \frac{\sum_{i=1}^{m} f_{ij}\cdot\boldsymbol{x}_i}{\sum_{i=1}^{m} f_{ij}} \quad (15)$$

With the new definition of $\mathbf{S}_w$, $\mathbf{S}_b$ and the optimization objective in Eq.(11), the projection matrix $\mathbf{P}$ can be solved by the following generalized eigenvalue problem:

$$\left(\left((1-\alpha)\mathbf{S}_w+\alpha\mathbf{I}_d\right)^{-1}\mathbf{S}_b\right)\boldsymbol{p}_i=\lambda_i\boldsymbol{p}_i \quad (16)$$

where λ_i and $\boldsymbol{p}_i$ represent the eigen value and the corresponding eigen vector respectively. According to the sorted result of eigen values in descending order, DRAW selects the top d' eigen values, while the corresponding eigen vectors form the projection matrix $\mathbf{P}$ for DRAW.

As shown in Eq.(13) and Eq.(14), the reliability of the labeling confidence directly affects the induction of projection matrix $\mathbf{P}$, at the same time, the identification of ground-truth label is a iterative process in gradual. Therefore, DRAW alternates between dimensionality reduction and candidate label disambiguation.

In the iterative process, on one hand, the projection matrix of DRAW is optimized by disambiguation-guided label confidences. On the other hand, the labeling confidences are disambiguated by leveraging the manifold structure in the label space and DRAW-induced feature space. Finally, projection matrix represents the instances with more compact feature vectors to alleviate the curse of dimensionality and decrease the risk of overfitting.

3.3 Recovering Label Confidence via Disambiguation

Based on the induced projection matrix $\mathbf{P}$, a new partial label data set $\mathcal{D}'=\{(\boldsymbol{x}'_i,S_i)\mid 1\le i\le m\}$ with d' dimensions in projected feature space is deduced from $\mathcal{D}=\{(\boldsymbol{x}_i,S_i)\mid 1\le i\le m\}$ with d dimensions in the original feature space, where $\boldsymbol{x}'_i=\mathbf{P}^\top\boldsymbol{x}_i$. As shown in Eq.(13) and Eq.(14), label confidence vector affect the quality of induced projection matrix in DRAW, thus, to acquire a more accurate projection matrix, the label confidence matrix is updated according to the manifold structure in the projected feature space and label space in each iteration.

Specifically, given the new partial label data set, we construct the similarity matrix via the weighted graph $\mathcal{G}=(V,E,\mathbf{W})$ to characterize the manifold structure in projected feature space and label space simultaneously, where $V=\{\boldsymbol{x}'_i\mid 1\le i\le m\}$ corresponds to the set of vertices, $E=\left\{(\boldsymbol{x}'_i,\boldsymbol{x}'_j)\mid \boldsymbol{x}'_i\in KNN(\boldsymbol{x}'_j),i\ne j\right\}$ describes the set of edges from $\boldsymbol{x}'_i$ to $\boldsymbol{x}'_j$ iff $\boldsymbol{x}'_i$ belongs to the k-nearest neighbors of $\boldsymbol{x}'_j$. Moreover, $\mathbf{W}\in\mathbb{R}^{m\times m}$ matches with the non-negative weight matrix, where $w_{ij}=0$ if $(\boldsymbol{x}'_i,\boldsymbol{x}'_j)\notin E$. For the new partial label data set, the optimal weight matrix can be induced as follows:

$$\begin{aligned}\min_{\mathbf{W}}\quad & \sum_{j=1}^{m}\|\,\boldsymbol{f}_j-\sum_{(\boldsymbol{x}'_i,\boldsymbol{x}'_j)\in E}w_{ij}\boldsymbol{f}_i\,\|_2^2\\ +\quad & \sum_{j=1}^{m}\|\,\boldsymbol{x}'_j-\sum_{(\boldsymbol{x}'_i,\boldsymbol{x}'_j)\in E}w_{ij}\boldsymbol{x}'_i\,\|_2^2\\ s.t.\quad & \mathbf{W}^\top\mathbf{1}_m=\mathbf{1}_m\end{aligned} \quad (17)$$

where $\boldsymbol{f}_i$ corresponds to the label confidence vector of the instance $\boldsymbol{x}'_i$.

For each instance, the similarity vector is independent with others, i.e. each column of similarity matrix $\mathbf{W}$ is independent individually. Therefore, we optimize the similarity vector one by one, and furthermore, the problem on instance $\boldsymbol{x}'_j$ can be rewritten as follows according to Eq.(17):

$$\begin{aligned}\min_{\mathbf{W}_{\cdot j}}\quad & \|\,\boldsymbol{f}_j-\sum_{(\boldsymbol{x}'_i,\boldsymbol{x}'_j)\in E}w_{ij}\boldsymbol{f}_i\,\|_2^2\\ +\quad & \|\,\boldsymbol{x}'_j-\sum_{(\boldsymbol{x}'_i,\boldsymbol{x}'_j)\in E}w_{ij}\boldsymbol{x}'_i\,\|_2^2\\ s.t.\quad & \mathbf{W}_{\cdot j}^\top\mathbf{1}_m=1\end{aligned} \quad (18)$$

where $\mathbf{W}_{\cdot j}$ denotes the elements in the j-th column, the value reflects reconstruction weights for instance $\boldsymbol{x}'_j$ and label confidence vector $\boldsymbol{f}_j$.

After inducing the weight matrix $\mathbf{W}$, we update the label confidence matrix according to the aggregation of k nearest neighbors via $\mathbf{W}$. Specifically, for label confidence vector $\boldsymbol{f}_i=[f_{i1},f_{i1},\cdots,f_{iq}]$, the updated label vector $\boldsymbol{f}'_i$ is calculated as follows:

$$\boldsymbol{f}'_i=\boldsymbol{f}_i+\sum_{\boldsymbol{x}'_j\in\mathcal{N}_k(\boldsymbol{x}'_i)}W_{ij}\boldsymbol{f}_j \quad (19)$$

where $\mathcal{N}_k$ represents the k nearest neighbors of instance $\boldsymbol{x}'_i$ in lower dimensional feature space, W_{ij} reflects the influence for $\boldsymbol{x}'_j$ on $\boldsymbol{x}'_i$. Then, the updated label confidence matrix $\boldsymbol{F}'$ is normalized, which ensures the sum of label confidence vector to 1 for each instance, while set the non-candidate labels to be zero.

$$\forall\,1\le i\le m,\ 1\le j\le q: F'_{ij}=\begin{cases}\frac{f'_{ij}}{\sum_{l_k\in S_i}f'_{ik}}, & \text{if } l_j\in S_i\\ 0, & \text{otherwise}\end{cases}$$

where $\mathbf{F}'$ corresponds to the updated label confidence matrix, and for each instance, the label confidence vector satisfies constraints $\forall\,1\le i\le m,\ 1\le j\le q, F'_{ij}\ge 0$ and $\sum_{j=1}^{q}F'_{ij}=1$.

3.4 Gradually Induced Discriminant Analysis

In the initial stage, as shown in Eq.(12), the label confidence vector is initialized with equal value for each label in candidate label set. Obviously, the label confidence vector cannot distinguish between the ground-truth label and false positive labels in candidate label set. Therefore, the supervised dimensionality reduction is easy to be misled by the false positive labels, which would make the training procedure

suffer from overfitting. Thus, we expect our DRAW rely less on this inaccurate supervision in the initial stage.

At the same time, it is worth noting that the identification of ground-truth label is a iterative process, and hence the reliability of supervision information increases step by step. In this procedure, the quality of label confidence directly affects the induction of projection matrix $\mathbf{P}$ as shown in Eq.(13) and Eq.(14). To better combine supervised and unsupervised dimensionality reduction, we employ a adaptive process to dynamically balance the contribution of PCA and LDA. In particular, DRAW adjust α with decay parameter s, i.e., $\alpha_t = \alpha_{t-1} \cdot s$, where s $(0 < s < 1)$ controls the speed to decrease the weight of PCA, and t is the number of iterations.

Obviously, with the process of iteration, the weight of LDA gradually increases while the weight of PCA decreases in calculating projection matrix. As the number of iterations increases, LDA will gradually dominate in the identification of projection matrix. A special case when the decay $s = 1$, the adaptive weight is ablated, as shown in Fig. 5, compared with $0 < s < 1$, on `Lost`, the partial label learning algorithms coupled with DRAW via adaptive weight achieves superior performance than the one with fixed value on s, which demonstrates that the adaptive weight fits in the characteristics of dimensionality reduction on partial label learning.

Table 1 characterizes the pseudo code of proposed algorithm DRAW, which firstly initializes the label confidence matrix $\mathbf{F}$ and instance matrix based on the partial label training data set (Steps 1-2). Then, the global mean vector is calculated via the instance matrix (Step 3). After that, an iterative procedure alternates between recovering label confidence (Steps 10-15) and dimensionality reduction (Steps 5-8) is conducted. In each iteration, DRAW induces the projection matrix according to the guideline of label confidence matrix, while the partial label disambiguation is executed by leveraging the manifold information in label space and lower-dimensional feature space via projection matrix $\mathbf{P}$. In this process, DRAW adjusts the weight of PCA and LDA with an adaptive weight decay scheme (Step 16). The iterative procedure terminates if the projection matrix $\mathbf{P}$ remains unchanged or the number of iterations reaches the maximum. Finally, the transformed partial label training data set $\mathcal{D}'$ with d'-dimensional features (Step 18) is utilized to induce the classification model $f : \mathcal{X}' \mapsto \mathcal{Y}$. Furthermore, for unseen instance $\boldsymbol{x}_i$, the model makes the prediction as $f(\mathbf{P}^\top \boldsymbol{x}_i)$.

4 EXPERIMENTS

4.1 Experimental Setup

4.1.1 Comparing methods

To evaluate the effectiveness of the proposed partial label dimensionality reduction approach, we integrate DRAW with state-of-the-art partial label learning algorithms. To the best of our knowledge, CENDA and DELIN are the only dimensionality reduction methods towards partial label learning. CENDA employs the Hilbert-Schmidt Independence Criterion to derive the projection matrix via confidence-based dependence maximization. DELIN adapts LDA to induce the projection matrix via the label confidence vector instead explicit label assignment, and hence it is also used as a adaptive LDA as baseline dimensionality reduction algorithm. Apart from DELIN and CENDA, since DRAW combines the strengths of PCA and LDA, we also include PCA as a baseline dimensionality reduction method for comparison purposes.

For each partial label learning algorithm $\mathcal{L}$, the coupled versions with different dimensionality reduction methods are denoted as $\mathcal{L}$-PCA, $\mathcal{L}$-DELIN, $\mathcal{L}$-CENDA and $\mathcal{L}$-DRAW respectively. Consequently, the classification model is derived from the lower-dimensional partial label training set, which is transformed via the projection matrix induced by PCA, DELIN, CENDA and DRAW individually. The performance of $\mathcal{L}$-DRAW is compared against $\mathcal{L}$-PCA, $\mathcal{L}$-DELIN, $\mathcal{L}$-CENDA and $\mathcal{L}$ respectively, which can verify the effectiveness of proposed dimensionality reduction approach DRAW in improving the generalization ability of partial label learning algorithms.

In this paper, we instantiate $\mathcal{L}$ using five well-established partial label learning algorithms, and their parameter configurations are set based on the recommendations provided in the corresponding literature.

- PL-KNN [66]: an averaging-based partial label learning algorithm, which makes the prediction for unseen instance by weighted voting on the candidate labels from kNN instances (parameter configuration: k=10).
- PL-SVM [67]: an identification-based partial label learning approach which induces the classification model by adapting the maximum margin to the candidate label set S_i and non-candidate label set $\mathcal{Y} \setminus S_i$ (parameter configuration: regularization coefficient pool with $\{10^{-3}, \ldots, 10^{3}\}$).
- IPAL [36]: a disambiguation-based partial label learning algorithm, which determines the valid label from candidate label set via label propagation on weighted graph based on kNN instances (parameter configuration: k=10, balancing coefficient $\alpha = 0.95$).
- SURE [68]: a self-training partial label learning algorithm, under proper constraints, which unifies the training of classification model and the identification of pseudo label jointly into one formulation. Partial label learning is transformed into a convex-concave optimization problem (parameter configuration: regularization coefficients λ = 0.3, β = 0.05).
- PL-AGGD [16]: a adaptive graph guided disambiguation algorithm towards partial label learning, which jointly performs the graph construction, classification model training and partial label disambiguation in a framework. Then, alternative optimization is utilized to tackle this problem (parameter configuration: $k = 10$ and balancing coefficients $\mu = 1$, $\gamma = 0.05$).

As shown in Table 1, for DRAW, α balances the contribution between supervised dimensionality reduction and unsupervised dimensionality reduction. At the same time, the decay parameter s is employed to control the pace in adjusting α. In this paper, considering that the ground-truth label is concealed in the candidate label set, while it is gradually identified during the disambiguation process,

TABLE 2
Characteristics of the synthetic partial label data sets.

Data Set	# Examples	# Features	# Class Labels	# False Positive Labels (r)	Task Domain
mediamill	2,854	120	10	$r = 1, 2, 3$	*video semantic detection* [55]
tmc2007	8,670	981	18	$r = 1, 2, 3$	*text anomaly detection* [56]
slashdot	3,142	1,079	19	$r = 1, 2, 3$	*text classification* [57]
amazon	1,500	1,326	50	$r = 1, 2, 3$	*authorship identification* [58]
DeliciousMIL	1,409	1,389	20	$r = 1, 2, 3$	*sentence labeling* [59]
bookmark	2,500	1,413	57	$r = 1, 2, 3$	*automatic tag suggestion* [60]
sports	9,120	1,738	19	$r = 1, 2, 3$	*human activity recognition* [61]
sector	6,412	6,104	105	$r = 1, 2, 3$	*text classification* [62]

TABLE 3
Characteristics of the real-world partial label data sets.

Data Set	# Examples	# Features	# Class Labels	average # Candidate Labels	Task Domain
FG-NET	1,002	262	78	7.48	*facial age estimation* [63]
Lost	1,122	108	16	2.23	*automatic face naming* [34]
MSRCv2	1,758	48	23	3.16	*object classification* [11]
Mirflickr	2,780	1,536	14	2.76	*web image classification* [64]
BirdSong	4,998	38	13	2.18	*bird song classification* [15]
Soccer Player	17,472	279	171	2.09	*automatic face naming* [65]
Yahoo! News	22,991	163	219	1.91	*automatic face naming* [9]

thus, the parameters are set as $\alpha = 0.5$ and $s = 0.95$ respectively.

4.1.2 Implementation

Our implementation is based on Matlab, and a windows server equipped with Intel i7-8700 CPU (@3.20GHz) and 40GB memory is used to support the experiments. For Oxford Flowers 102, Flowers recognition and Caltech-101, we employ GoogleNet as the backbone neural network to extract features. In the following subsections, for each data set, ten-fold cross-validation is performed, while the mean and standard deviation of prediction accuracy are recorded.

4.2 Synthetic Data Sets

Following the widely-used protocal in partial label learning studies [69], [70], the synthetic partial label data sets are generated from multi-class data sets via different configuration. Specifically, r is a parameter to control the number of false positive labels added in the candidate label set S_i. For each multi-class instance $(\boldsymbol{x}_i, y_i)$, it is transformed into a partial label instance $(\boldsymbol{x}_i, S_i)$ by concealing the ground-truth label y_i into r false positive labels P^r, which are randomly selected, i.e. $S_i = y_i \cup P^r$ and $\mid S_i \mid = r + 1$.

Table 2 summarizes the characteristics of synthetic partial label data sets used in this paper, where the r is set as $\{1, 2, 3\}$ respectively. Accordingly, the detailed experimental results over synthetic data sets on each comparing algorithms are reported in Table 7, Table 8 and Table 9. These tables are shown in the number of false positive labels in ascending order ($r \in \{1, 2, 3\}$).

For each partial label learning algorithm, $\mathcal{L} \in \{$PL-KNN, PL-SVM, IPAL, SURE, PL-AGGD$\}$, $\mathcal{L}$-DRAW is compared against $\mathcal{L}$-CENDA, $\mathcal{L}$-DELIN, $\mathcal{L}$-PCA and $\mathcal{L}$, and the best classification performance is shown in boldface. Fig. 3 presents the classification accuracy of each partial label algorithm before and after employing four dimensionality reduction methods (PCA, DELIN, CENDA and DRAW) on `sports` and `DeliciousMIL`. Moreover, pairwise t-test at 0.05 significance level is conducted to show whether the performance difference between two comparison methods is significant in statistics, which includes $\mathcal{L}$-DRAW and $\mathcal{L}$-CENDA, $\mathcal{L}$-DRAW and $\mathcal{L}$-DELIN. The results are recorded in Table 6, which shows the win/tie/loss counts between $\mathcal{L}$-DRAW and different dimensionality reduction methods on each partial label learning algorithm. Based on these comparative results, the following observations can be concluded:

- Compared with partial label learning algorithms $\mathcal{L}$, across all the 120 statistical comparison cases (8 data sets $\times$ 3 configurations $\times$ 5 algorithms), $\mathcal{L}$-DRAW achieves superior performance against $\mathcal{L}$ in all cases after employing the proposed dimensionality reduction approach DRAW.
- Compared with the classification results achieved via partial label learning algorithms coupled with a single PCA, $\mathcal{L}$-DRAW achieves better classification performance with a pretty large margin over 100% cases.
- Compared with the existing partial label dimensionality reduction method CENDA, $\mathcal{L}$-DRAW achieves better classification performance against $\mathcal{L}$-CENDA

TABLE 4
Classification accuracy (mean ± std) of each comparing algorithm on real-world partial label data sets. For partial label learning algorithm $\mathcal{L} \in \{$PL-KNN, PL-SVM, IPAL, SURE, PL-AGGD$\}$, the performance of $\mathcal{L}$-DRAW ,$\mathcal{L}$-DELIN and $\mathcal{L}$-CENDA are compared against that of $\mathcal{L}$, where the best performance is shown in bold face.

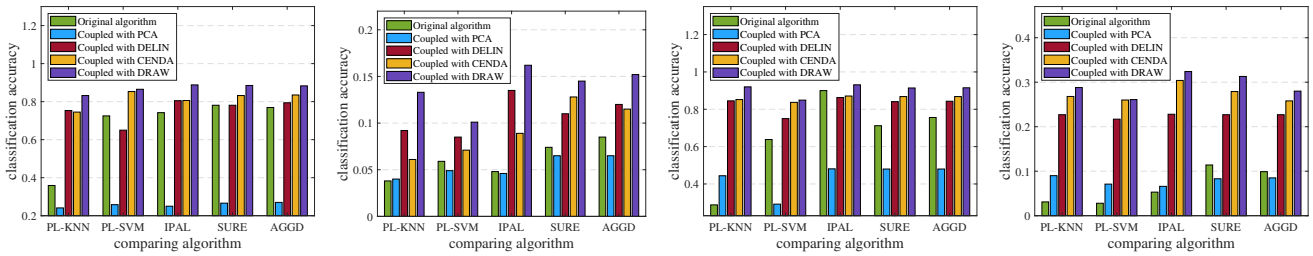
Comparing Algorithm	Data Set						
	Lost	FG-NET	Mirflickr	Yahoo! News	Soccer Player	BirdSong	MSRCv2
PL-KNN	0.359±0.032	0.038±0.011	**0.499±0.023**	0.413±0.005	0.493±0.004	**0.648±0.017**	0.437±0.022
PL-KNN-PCA	0.241±0.033	0.040±0.021	0.485±0.024	0.413±0.005	0.493±0.003	0.642±0.015	0.440±0.026
PL-KNN-DELIN	0.753±0.029	0.092±0.016	0.496±0.029	0.426±0.007	0.495±0.005	0.574±0.020	0.417±0.019
PL-KNN-CENDA	0.745±0.017	0.061±0.019	0.492±0.025	0.340±0.007	0.492±0.003	0.630±0.023	0.433±0.017
PL-KNN-DRAW	**0.832±0.018**	**0.133±0.031**	0.484±0.018	**0.443±0.006**	**0.495±0.003**	0.640±0.012	**0.445±0.021**
PL-SVM	0.725±0.048	0.059±0.025	0.496±0.087	0.517±0.009	0.402±0.050	0.479±0.042	0.321±0.030
PL-SVM-PCA	0.258±0.037	0.049±0.023	0.481±0.040	0.517±0.008	0.373±0.020	0.484±0.046	0.315±0.051
PL-SVM-DELIN	0.650±0.065	0.085±0.017	0.546±0.126	0.451±0.013	0.370±0.011	0.377±0.053	0.356±0.033
PL-SVM-CENDA	0.853±0.027	0.071±0.030	**0.562±0.096**	**0.592±0.011**	0.433±0.016	0.469±0.011	**0.412±0.032**
PL-SVM-DRAW	**0.865±0.027**	**0.101±0.026**	0.561±0.071	0.465±0.010	**0.471±0.013**	**0.564±0.042**	0.373±0.070
IPAL	0.742±0.050	0.048±0.011	**0.536±0.025**	0.671±0.007	0.550±0.007	**0.710±0.012**	0.539±0.035
IPAL-PCA	0.250±0.030	0.046±0.027	0.478±0.025	0.671±0.007	0.541±0.008	0.680±0.019	0.509±0.034
IPAL-DELIN	0.805±0.034	0.135±0.042	0.496±0.028	0.671±0.007	0.557±0.011	0.617±0.024	0.500±0.028
IPAL-CENDA	0.806±0.023	0.089±0.019	0.418±0.010	0.646±0.008	0.551±0.008	0.683±0.011	0.519±0.030
IPAL-DRAW	**0.888±0.030**	**0.162±0.029**	0.484±0.026	**0.675±0.007**	**0.562±0.008**	0.706±0.014	**0.540±0.016**
SURE	0.781±0.030	0.074±0.025	0.670±0.021	0.638±0.007	0.534±0.005	**0.745±0.026**	0.480±0.024
SURE-PCA	0.266±0.025	0.065±0.031	0.499±0.024	0.638±0.007	0.528±0.006	0.691±0.022	0.461±0.025
SURE-DELIN	0.781±0.028	0.110±0.021	0.688±0.028	0.642±0.008	0.541±0.006	0.640±0.017	0.476±0.028
SURE-CENDA	0.832±0.030	0.128±0.032	**0.707±0.044**	0.624±0.008	0.536±0.004	0.682±0.017	0.490±0.036
SURE-DRAW	**0.885±0.024**	**0.145±0.034**	0.687±0.028	**0.645±0.009**	**0.541±0.006**	0.724±0.022	**0.490±0.023**
PL-AGGD	0.769±0.033	0.085±0.029	0.665±0.018	0.657±0.009	0.544±0.005	**0.735±0.018**	0.497±0.028
PL-AGGD-PCA	0.270±0.018	0.065±0.022	0.526±0.020	0.657±0.009	0.533±0.009	0.694±0.017	0.484±0.022
PL-AGGD-DELIN	0.794±0.029	0.120±0.032	0.575±0.047	0.659±0.008	0.545±0.005	0.645±0.017	0.486±0.018
PL-AGGD-CENDA	0.835±0.030	0.115±0.030	0.554±0.048	0.640±0.010	0.539±0.005	0.695±0.017	0.486±0.022
PL-AGGD-DRAW	**0.883±0.027**	**0.152±0.039**	**0.673±0.024**	**0.663±0.008**	**0.550±0.007**	0.712±0.012	**0.505±0.030**

TABLE 5
Win/tie/loss statistic (pairwise t-test at 0.05 significance level) between $\mathcal{L}$-DRAW, $\mathcal{L}$-DELIN and $\mathcal{L}$-CENDA on each real-world partial label data set.

Data Set	$\mathcal{L}$-DRAW against $\mathcal{L}$-DELIN					$\mathcal{L}$-DRAW against $\mathcal{L}$-CENDA				
	$\mathcal{L}$=PL-KNN	$\mathcal{L}$= PL-SVM	$\mathcal{L}$=IPAL	$\mathcal{L}$=SURE	$\mathcal{L}$=PL-AGGD	$\mathcal{L}$= PL-KNN	$\mathcal{L}$= PL-SVM	$\mathcal{L}$=IPAL	$\mathcal{L}$=SURE	$\mathcal{L}$=PL-AGGD
Lost	win	win	win	win	win	win	win	win	win	win
FG-NET	win	loss	win	win	win	win	win	win	win	win
Mirflickr	loss	win	loss	loss	win	loss	loss	win	loss	win
Yahoo! News	win	win	win	win	win	win	loss	win	win	win
Soccer Player	win	win	win	loss	win	win	win	win	win	win
BirdSong	win	loss	win	win	win	win	win	win	win	win
MSRCv2	win	tie	win	win	win	win	loss	win	loss	win
Total	**6/0/1**	**4/1/2**	**6/0/1**	**5/0/2**	**6/0/1**	**6/0/1**	**4/0/3**	**7/0/0**	**5/0/2**	**7/0/0**

across 101 out of 120 cases.

- Compared with DELIN, $\mathcal{L}$-DRAW achieves better performance than $\mathcal{L}$-DELIN in 90 out of 120 cases. Furthermore, since DELIN achieves dimensionality reduction with a single LDA, the comparison results indicates that DRAW is more consistent with the characteristics of partial label learning. The supervised dimensionality reduction and unsupervised dimensionality reduction is complementary.
- Based on the observation in Table 7, Table 8 and Table 9, on `mediamill`, the improvement on classification accuracy is the mildest in synthetic data sets, while the number of feature in `mediamill` is the smallest. For the other data sets, PL-KNN couple with DRAW achieves performance improvement by **0.3** in 17 cases out of 21 cases (7 data sets $\times$ 3 configurations).
- On the two data sets with least number examples, `amazon` and `DeliciousMIL`, the classification accuracy has been improved with DRAW by more than **0.2**, **0.2** and **0.1** for $r = 1, 2$ and 3 in `DelicsectorL`, respectively. And **0.5**, **0.35** and **0.29** improvements arises on `amazon`. These results indicate that the clas-



(a) Lost (b) FG-NET (c) sports ($r = 2$) (d) DeliciousMIL ($r = 2$)

Fig. 3. Comparison of the classification accuracy of each partial label learning algorithm on real-world and synthetic partial label data sets before (*green* bar) and after employing PCA (*blue* bar), DELIN (*red* bar), CENDA (*yellow* bar) and DRAW (*purple* bar).

TABLE 6
Win/tie/loss counts (pairwise t-test at 0.05 significance level) between $\mathcal{L}$-DRAW, $\mathcal{L}$-DELIN and $\mathcal{L}$-CENDA in terms of different number of false positive labels ($r = 1, 2, 3$) on synthetic partial label data sets.

Data Set	$\mathcal{L}$-DRAW against $\mathcal{L}$-DELIN					$\mathcal{L}$-DRAW against $\mathcal{L}$-CENDA				
	$\mathcal{L}$=PL-KNN	$\mathcal{L}$= PL-SVM	$\mathcal{L}$=IPAL	$\mathcal{L}$=SURE	$\mathcal{L}$=PL-AGGD	$\mathcal{L}$= PL-KNN	$\mathcal{L}$= PL-SVM	$\mathcal{L}$=IPAL	$\mathcal{L}$=SURE	$\mathcal{L}$=PL-AGGD
$r = 1$	6/0/2	8/0/0	7/0/1	6/0/2	6/0/2	5/0/3	5/0/3	6/0/2	5/0/3	5/0/3
$r = 2$	7/0/1	7/0/1	7/0/1	7/0/1	7/0/1	8/0/0	5/0/3	7/0/1	6/0/2	6/0/2
$r = 3$	7/0/1	5/0/3	7/0/1	7/0/1	7/0/1	7/0/1	5/0/3	7/0/1	6/0/2	7/0/1
Total	**20/0/4**	**20/0/4**	**21/0/3**	**20/0/4**	**20/0/4**	**20/0/4**	**15/0/9**	**20/0/4**	**17/0/7**	**18/0/6**

sification performance can benefit from DRAW even if the number of training examples is insufficient.

- For high dimensional datasets, such as `amazon`, `DeliciousMIL`, `bookmark`, `sports` and `sector`, the number of features in those data sets is more than 1,300. For PL-KNN and PL-SVM, the classification performance is significantly improved by DRAW by **0.2** in 22 cases among 30 cases (2 comparing algorithms $\times$ 5 datasets $\times$ 3 configurations).

4.3 Real-World Data Sets

A number of real-world partial label data sets have been collected from different tasks and domains, which includes `FG-NET` for facial age estimation [63], `Lost`, `Soccer Player` and `Yahoo! News` for automatic face naming [9], [34], [65], `MSRCv2` for object classification [11], `Mirflickr` for web image classification [64], `BirdSong` for bird song classification [15].

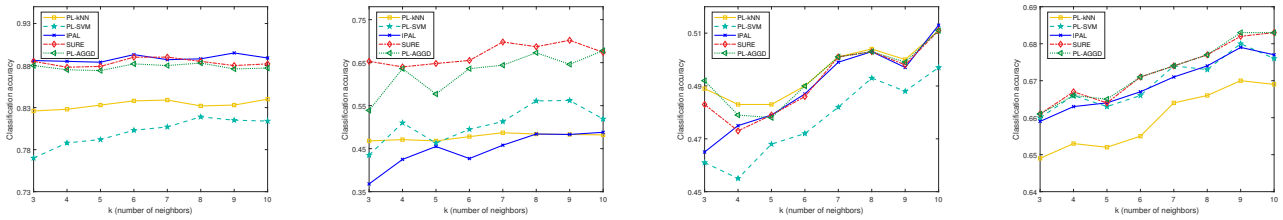
Specifically, for the task of *facial age estimation*, a human face with landmarks is regarded as instance, accordingly, the ages annotated by crowd-sourced labelers consist of the candidate label set. In *automatic face naming*, faces cropped from an image or video frame are defined as instances while names extracted from the associated captions or subtitles constitute the candidate label set. In the task of *object classification*, image segmentation serves as the instances, and the objects that appear in the same image compose the matching candidate label set. In *bird song classification*, singing syllables of the birds are represented as instances, while the bird species that sing jointly in 10-second duration constitute the candidate label set. In *web image classification*, web image is represented as a instance while annotations extracted from the web environment serve as the candidate label set. The characteristics of real-world partial label data sets is summarized in Table 3.

The classification accuracy over real-world data sets on each comparing algorithms is reported in Table 4, with the best classification performance shown in boldface. Additionally, Fig. 3 illustrates the classification accuracy of the partial label algorithm before and after employing four dimensionality reduction methods PCA, DELIN, CENDA and DRAW on `Lost` and `FG-NET` datasets. The pairwise t-test at 0.05 significance level is conducted to validate whether the classification performance difference is significant in statistics for two comparison methods, i.e. $\mathcal{L}$-DRAW and $\mathcal{L}$-DELIN, $\mathcal{L}$-DRAW and $\mathcal{L}$-CENDA. The win/tie/loss statistics are reported in Table 5. According to the reported results on each real-world data sets, the following observations can be concluded:

- Compared with partial label learning algorithms $\mathcal{L}$, across 35 statistical comparisons (7 data sets $\times$ 5 algorithms), the prediction accuracy has been significantly improved by employing DRAW in 28 cases.
- For the existing partial label dimensionality reduction methods $\mathcal{L}$-DELIN and $\mathcal{L}$-CENDA, $\mathcal{L}$-DRAW achieves better performance improvement in 27 cases against $\mathcal{L}$-DELIN, and there is only one tie on `MSRCv2`. For $\mathcal{L}$-CENDA, $\mathcal{L}$-DRAW achieves significant performance improvement in 29 cases in pairwise t-test at 0.05 significance level.
- As shown in Table 4, compared with the classification results achieved via partial label learning algorithms coupled with a single PCA, $\mathcal{L}$-DRAW achieves better classification performance in almost all cases.
- As shown in Fig. 3(b), on `FG-NET` dataset, the performance improvement of DRAW is impressive than $\mathcal{L}$, $\mathcal{L}$-PCA, $\mathcal{L}$-DELIN and $\mathcal{L}$-CENDA. From Table 3,

TABLE 7
Classification accuracy (mean ± std) of each comparing algorithm on controlled synthetic data sets with varying number of false positive labels $r = 1$ respectively. For partial label learning algorithm $\mathcal{L} \in \{$PL-KNN, PL-SVM, IPAL, SURE, PL-AGGD$\}$, the performance of $\mathcal{L}$-DRAW ,$\mathcal{L}$-DELIN and $\mathcal{L}$-CENDA are compared against that of $\mathcal{L}$, where the best performance is shown in bold face.

Comparing Algorithm	Data Set							
	mediamill	tmc2007	slashdot	amazon	DeliciousMIL	bookmark	sports	sector
	$r = 1$ (one false positive label)							
PL-KNN	0.635±0.018	0.398±0.016	0.163±0.008	0.024±0.006	0.038±0.038	0.167±0.030	0.288±0.005	0.015±0.002
PL-KNN-PCA	0.631±0.015	0.604±0.010	0.490±0.027	0.035±0.015	0.073±0.025	0.228±0.020	0.446±0.012	0.025±0.007
PL-KNN-DELIN	0.682±0.018	0.699±0.015	0.551±0.020	0.647±0.024	**0.436±0.035**	**0.487±0.032**	0.842±0.010	0.522±0.017
PL-KNN-CENDA	**0.707±0.027**	0.746±0.016	0.740±0.024	0.650±0.041	0.420±0.035	0.422±0.025	0.900±0.011	0.549±0.016
PL-KNN-DRAW	0.706±0.026	**0.748±0.016**	**0.746±0.018**	**0.701±0.050**	0.387±0.035	0.480±0.034	**0.939±0.007**	**0.557±0.018**
PL-SVM	0.478±0.055	0.605±0.140	0.592±0.024	0.107±0.025	0.033±0.016	0.285±0.018	0.679±0.006	0.073±0.014
PL-SVM-PCA	0.475±0.040	0.578±0.015	0.516±0.018	0.054±0.014	0.051±0.009	0.289±0.020	0.283±0.028	0.018±0.005
PL-SVM-DELIN	0.595±0.037	0.697±0.010	0.572±0.024	0.642±0.024	0.378±0.026	0.434±0.074	0.766±0.021	0.506±0.010
PL-SVM-CENDA	**0.627±0.040**	**0.758±0.014**	**0.758±0.022**	0.649±0.041	0.433±0.048	0.486±0.027	0.873±0.014	0.550±0.015
PL-SVM-DRAW	0.615±0.038	0.726±0.009	0.750±0.022	**0.698±0.051**	**0.446±0.054**	**0.507±0.045**	**0.882±0.011**	**0.561±0.018**
IPAL	0.646±0.035	0.592±0.021	0.421±0.019	0.106±0.026	0.063±0.018	0.316±0.015	0.905±0.007	0.145±0.016
IPAL-PCA	0.633±0.031	0.577±0.013	0.497±0.021	0.069±0.019	0.070±0.021	0.296±0.011	0.481±0.011	0.078±0.007
IPAL-DELIN	0.655±0.018	0.660±0.016	0.586±0.020	0.647±0.023	0.435±0.035	**0.506±0.029**	0.861±0.007	0.521±0.017
IPAL-CENDA	0.665±0.027	**0.719±0.016**	0.744±0.023	0.653±0.039	**0.489±0.043**	0.435±0.030	0.917±0.011	0.550±0.016
IPAL-DRAW	**0.673±0.025**	0.716±0.015	**0.755±0.015**	**0.704±0.050**	0.442±0.048	0.499±0.031	**0.958±0.006**	**0.563±0.021**
SURE	0.695±0.026	0.647±0.016	0.582±0.017	0.165±0.028	0.117±0.012	0.393±0.015	0.753±0.009	0.124±0.016
SURE-PCA	0.669±0.018	0.639±0.015	0.557±0.018	0.102±0.031	0.097±0.023	0.308±0.019	0.480±0.012	0.107±0.012
SURE-DELIN	0.712±0.022	0.716±0.015	0.696±0.027	0.647±0.024	0.433±0.031	**0.525±0.029**	0.843±0.008	0.522±0.017
SURE-CENDA	0.716±0.024	**0.764±0.012**	**0.766±0.021**	0.651±0.040	**0.503±0.036**	0.498±0.031	0.914±0.011	0.550±0.016
SURE-DRAW	**0.722±0.021**	0.755±0.016	0.762±0.012	**0.703±0.050**	0.428±0.066	0.521±0.032	**0.944±0.007**	**0.558±0.020**
PL-AGGD	0.697±0.026	0.650±0.015	0.575±0.015	0.157±0.032	0.120±0.022	0.390±0.013	0.773±0.008	0.121±0.015
PL-AGGD-PCA	0.667±0.014	0.640±0.014	0.554±0.020	0.098±0.033	0.102±0.022	0.307±0.016	0.484±0.013	0.104±0.011
PL-AGGD-DELIN	0.707±0.021	0.713±0.016	0.694±0.029	0.647±0.024	0.433±0.031	**0.525±0.030**	0.845±0.008	0.522±0.017
PL-AGGD-CENDA	0.716±0.024	**0.761±0.010**	**0.766±0.021**	0.651±0.039	**0.437±0.063**	0.498±0.032	0.916±0.011	0.551±0.016
PL-AGGD-DRAW	**0.721±0.023**	0.755±0.016	0.763±0.012	**0.702±0.051**	0.362±0.035	0.521±0.034	**0.946±0.007**	**0.557±0.018**



(a) Lost (b) Mirflickr (c) amazon ($r = 2$) (d) slashdot ($r = 2$)

Fig. 4. Parameter sensitivity analysis for $\mathcal{L}$-DRAW ($\mathcal{L} \in \{$PL-KNN, PL-SVM, IPAL, SURE, PL-AGGD$\}$), classification accuracy changes as the number of nearest neighbors used for canadidate label set disambiguation (i.e., k) increases from 3 to 10 with step-size 1. (a) real-world data set `Lost`. (b) real-world data set `Mirflickr`. (c) synthetic data set `amazon` ($r = 2$). (d) synthetic data set `slashdot` ($r = 2$).

`FG-NET` is a challenging dataset, which holds least number of examples but the largest average number in candidate labels. Therefore, the result indicates that DRAW can significantly improve the performance in difficult circumstance with insufficient examples and high rate of false positive labels.

- As shown in Fig. 3(a), among 4 dimensionality reduction methods PCA, DELIN, CENDA and DRAW, DRAW achieves superior performance in all cases on `Lost` (5 partial label learning algorithms). For the other real-world partial label datasets, DRAW achieves superior superior or at least statistically comparable performance against DELIN and CENDA in most cases.

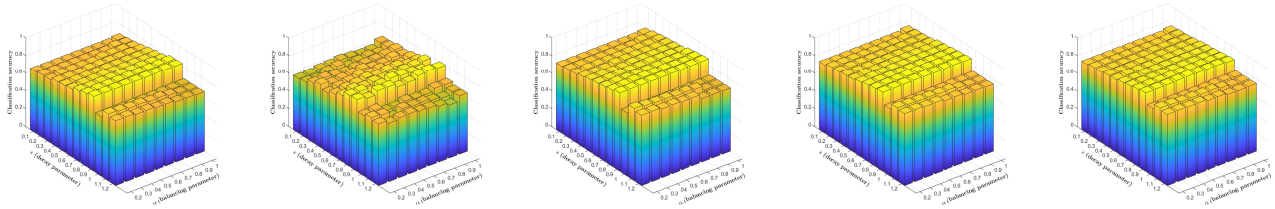
4.4 Sensitivity Analysis

As shown in Table 1, d' plays a crucial parameter to control the number of retained features after employing dimensionality reduction. Following the common practice for LDA in multi-class classification, we compare the classification accuracy of partial label learning algorithms after employing DRAW with varying d'. The value of d' is set as a certain proportion of $min(d, q-1)$, which is denoted as $d' = \lceil ratio \cdot \min(d, q-1) \rceil$, where the value of $ratio$ reflects the remained proportion of original features.

Table 10 presents the classification accuracy of partial label learning algorithms coupled with DRAW on real-world data sets with varying value of $ratio$ from 0.5 to 1 with an interval of 0.1. The best result of each cell is highlighted in boldface. As shown in Table 10, the classification per-

TABLE 8
Classification accuracy (mean ± std) of each comparing algorithm on controlled synthetic data sets with varying number of false positive labels $r = 2$ respectively. For partial label learning algorithm $\mathcal{L} \in \{$PL-KNN, PL-SVM, IPAL, SURE, PL-AGGD$\}$, the performance of $\mathcal{L}$-DRAW ,$\mathcal{L}$-DELIN and $\mathcal{L}$-CENDA are compared against that of $\mathcal{L}$, where the best performance is shown in bold face.

Comparing Algorithm	Data Set							
	mediamill	tmc2007	slashdot	amazon	DeliciousMIL	bookmark	sports	sector
	$r = 2$ (two false positive label)							
PL-KNN	0.622±0.020	0.382±0.013	0.159±0.008	0.024±0.010	0.031±0.020	0.157±0.014	0.288±0.003	0.013±0.001
PL-KNN-PCA	0.609±0.018	0.585±0.009	0.471±0.027	0.027±0.013	0.090±0.030	0.233±0.014	0.444±0.008	0.022±0.004
PL-KNN-DELIN	0.663±0.026	0.683±0.014	0.588±0.019	0.483±0.049	0.227±0.032	**0.466±0.026**	0.845±0.014	0.394±0.012
PL-KNN-CENDA	0.673±0.021	0.727±0.011	0.656±0.029	0.493±0.045	0.268±0.038	0.359±0.023	0.852±0.011	0.386±0.026
PL-KNN-DRAW	**0.685±0.026**	**0.729±0.011**	**0.666±0.028**	**0.504±0.027**	**0.288±0.041**	0.428±0.021	**0.920±0.010**	**0.402±0.017**
PL-SVM	0.503±0.031	0.637±0.012	0.576±0.023	0.067±0.019	0.028±0.013	0.254±0.020	0.638±0.012	0.053±0.010
PL-SVM-PCA	0.458±0.031	0.567±0.024	0.512±0.025	0.039±0.018	0.071±0.020	0.288±0.020	0.292±0.017	0.018±0.005
PL-SVM-DELIN	0.576±0.040	0.681±0.019	0.637±0.023	0.481±0.052	0.217±0.035	0.423±0.027	0.750±0.026	0.389±0.013
PL-SVM-CENDA	0.599±0.033	**0.740±0.010**	**0.681±0.029**	0.491±0.045	0.260±0.033	0.411±0.035	0.837±0.008	0.385±0.027
PL-SVM-DRAW	**0.593±0.035**	0.703±0.015	0.668±0.036	**0.492±0.028**	**0.261±0.036**	**0.427±0.026**	**0.849±0.008**	**0.397±0.015**
IPAL	0.591±0.027	0.579±0.014	0.406±0.017	0.099±0.019	0.053±0.011	0.307±0.022	0.900±0.006	0.137±0.014
IPAL-PCA	0.571±0.028	0.553±0.014	0.482±0.032	0.053±0.016	0.066±0.023	0.291±0.018	0.481±0.009	0.068±0.010
IPAL-DELIN	0.623±0.025	0.653±0.011	0.595±0.023	0.484±0.050	0.228±0.032	**0.484±0.017**	0.863±0.015	0.394±0.012
IPAL-CENDA	0.629±0.029	**0.698±0.010**	0.664±0.035	0.491±0.047	0.304±0.032	0.371±0.024	0.871±0.006	0.389±0.025
IPAL-DRAW	**0.644±0.023**	0.693±0.011	**0.674±0.030**	**0.503±0.028**	**0.324±0.041**	0.447±0.034	**0.931±0.009**	**0.403±0.018**
SURE	0.689±0.025	0.639±0.006	0.573±0.021	0.109±0.023	0.114±0.005	0.373±0.02	0.712±0.015	0.111±0.012
SURE-PCA	0.666±0.018	0.629±0.007	0.544±0.034	0.072±0.011	0.083±0.024	0.314±0.018	0.48-±0.012	0.094±0.008
SURE-DELIN	0.694±0.020	0.706±0.012	0.645±0.026	0.486±0.049	0.227±0.032	**0.510±0.014**	0.841±0.012	0.394±0.012
SURE-CENDA	0.710±0.015	**0.742±0.013**	**0.681±0.028**	0.491±0.045	0.279±0.026	0.419±0.027	0.868±0.008	0.388±0.025
SURE-DRAW	**0.712±0.025**	0.740±0.010	0.677±0.030	**0.503±0.028**	**0.313±0.027**	0.470±0.030	**0.914±0.009**	**0.401±0.016**
PL-AGGD	0.687±0.023	0.637±0.007	0.564±0.017	0.131±0.024	0.099±0.020	0.372±0.019	0.756±0.010	0.106±0.013
PL-AGGD-PCA	0.666±0.020	0.632±0.008	0.548±0.031	0.076±0.011	0.085±0.018	0.309±0.020	0.480±0.010	0.087±0.011
PL-AGGD-DELIN	0.691±0.021	0.704±0.012	0.642±0.024	0.485±0.051	0.227±0.032	**0.510±0.018**	0.843±0.012	0.394±0.012
PL-AGGD-CENDA	0.706±0.019	**0.743±0.013**	**0.681±0.030**	0.492±0.046	0.258±0.031	0.421±0.029	0.868±0.008	0.386±0.025
PL-AGGD-DRAW	**0.714±0.026**	0.739±0.010	0.677±0.031	**0.503±0.027**	**0.280±0.041**	0.470±0.030	**0.915±0.010**	**0.402±0.017**



(a) PL-KNN (b) PL-SVM (c) IPAL (d) SURE (e) PL-AGGD

Fig. 5. Parameter sensitivity analysis for $\mathcal{L}$-DRAW ($\mathcal{L} \in \{$PL-KNN, PL-SVM, IPAL, SURE, PL-AGGD$\}$). (a) Classification accuracy of PL-KNN on `Lost` by varying α and s. (b) Classification accuracy of PL-SVM on `Lost` by varying α and s. (c) Classification accuracy of IPAL on `Lost` by varying α and s. (d) Classification accuracy of SURE on `Lost` by varying α and s. (e) Classification accuracy of PL-AGGD on `Lost` by varying α and s.

formance of each partial label learning algorithm, when coupled with DRAW, exhibits moderate fluctuations as the remaining proportion of the original feature changes. Obviously, there is no single value for the remained proportion $ratio$ that consistently achieves the best classification performance in every case. Therefore, to further analyze performance improvement, fine-tuning the value of the remained proportion of the original feature becomes necessary for different datasets and partial label learning algorithms.

Furthermore, k (the number of nearest neighbors), α (trade-off parameter between LDA and PCA in Eq.(11)) and s (decay parameter) also serve as important parameters for DRAW. Coupled with DRAW, Fig. 4 illustrates how the classification accuracy of each partial label learning algorithm changes respectively as k increases from 3 to 10 with interval 1. As shown in Fig. 4, on the real-world data sets `Lost` and `Mirflickr`, synthetic data sets including `amazon` and `slashdot` with setting $r = 2$, the classification accuracy of each partial label learning algorithm coupled with DRAW is relatively stable as the value of k varies.

In addition to k, α (balancing parameter in Eq.(11)) increases from 0.2 to 1 with step-size 0.1 and s (decay parameter) increases from 0.1 to 1.2 with interval 0.1, where s controls the speed to decrease the weight of PCA in optimizing the projection matrix. Accordingly, when the values of s is close to 0, the descending speed of weight on PCA is rapid, with the increase of s, the speed slows down. As shown in Fig. 5, on real-world `Lost`, for different partial label algorithms, when the value of s is less than 1, $s \in \{0.1, 0.2, \cdots, 0.9\}$, compared with rapid weight

TABLE 9
Classification accuracy (mean ± std) of each comparing algorithm on controlled synthetic data sets with varying number of false positive labels $r = 3$ respectively. For partial label learning algorithm $\mathcal{L} \in \{$PL-KNN, PL-SVM, IPAL, SURE, PL-AGGD$\}$, the performance of $\mathcal{L}$-DRAW ,$\mathcal{L}$-DELIN and $\mathcal{L}$-CENDA are compared against that of $\mathcal{L}$, where the best performance is shown in bold face.

Comparing Algorithm	Data Set							
	mediamill	tmc2007	slashdot	amazon	DeliciousMIL	bookmark	sports	sector
	$r = 3$ (three false positive label)							
PL-KNN	0.600±0.021	0.370±0.009	0.164±0.009	0.028±0.008	0.043±0.005	0.141±0.010	0.292±0.004	0.016±0.003
PL-KNN-PCA	0.596±0.031	0.565±0.021	0.458±0.035	0.025±0.013	0.080±0.020	0.233±0.012	0.433±0.009	0.024±0.006
PL-KNN-DELIN	0.645±0.016	0.661±0.021	0.572±0.024	0.367±0.035	0.157±0.027	**0.420±0.015**	0.822±0.009	0.292±0.019
PL-KNN-CENDA	0.661±0.031	**0.704±0.018**	0.586±0.032	0.375±0.044	0.227±0.036	0.341±0.024	0.837±0.009	0.301±0.016
PL-KNN-DRAW	**0.666±0.017**	0.696±0.021	**0.605±0.025**	**0.408±0.027**	**0.250±0.039**	0.377±0.015	**0.898±0.010**	**0.305±0.015**
PL-SVM	0.459±0.034	0.626±0.018	0.563±0.026	0.053±0.015	0.028±0.012	0.252±0.012	0.606±0.011	0.048±0.008
PL-SVM-PCA	0.466±0.024	0.566±0.010	0.514±0.013	0.031±0.015	0.061±0.016	0.281±0.017	0.299±0.020	0.014±0.003
PL-SVM-DELIN	0.550±0.027	0.649±0.040	0.603±0.022	0.327±0.043	0.121±0.025	0.356±0.035	0.678±0.027	0.272±0.022
PL-SVM-CENDA	**0.584±0.024**	**0.719±0.016**	0.601±0.026	0.377±0.041	0.195±0.031	0.373±0.033	0.811±0.013	**0.299±0.018**
PL-SVM-DRAW	0.564±0.010	0.667±0.024	**0.606±0.030**	**0.409±0.033**	**0.212±0.034**	**0.388±0.021**	**0.819±0.018**	0.294±0.013
IPAL	0.532±0.027	0.560±0.017	0.363±0.024	0.088±0.009	0.043±0.016	0.294±0.020	0.890±0.007	0.140±0.010
IPAL-PCA	0.508±0.030	0.536±0.023	0.465±0.023	0.058±0.018	0.055±0.014	0.285±0.012	0.467±0.011	0.062±0.007
IPAL-DELIN	0.590±0.025	0.640±0.020	0.572±0.029	0.368±0.035	0.157±0.027	**0.426±0.025**	0.840±0.009	0.296±0.020
IPAL-CENDA	0.591±0.022	**0.676±0.015**	0.576±0.029	0.377±0.042	0.250±0.041	0.325±0.021	0.858±0.004	0.306±0.019
IPAL-DRAW	**0.619±0.031**	0.672±0.016	**0.597±0.025**	**0.407±0.028**	**0.271±0.034**	0.394±0.020	**0.910±0.008**	**0.312±0.014**
SURE	0.667±0.020	0.628±0.016	0.540±0.031	0.079±0.017	0.116±0.006	0.371±0.020	0.671±0.013	0.100±0.009
SURE-PCA	0.645±0.020	0.625±0.019	0.537±0.021	0.077±0.020	0.095±0.025	0.312±0.012	0.456±0.015	0.075±0.005
SURE-DELIN	0.692±0.015	0.691±0.017	0.604±0.028	0.369±0.036	0.157±0.027	**0.466±0.014**	0.817±0.010	0.296±0.021
SURE-CENDA	0.697±0.024	**0.725±0.017**	0.605±0.027	0.374±0.046	**0.228±0.025**	0.392±0.024	0.850±0.007	0.301±0.016
SURE-DRAW	**0.703±0.027**	0.718±0.018	**0.615±0.022**	**0.410±0.025**	**0.228±0.042**	0.418±0.017	**0.893±0.011**	**0.305±0.012**
PL-AGGD	0.674±0.021	0.633±0.01	0.533±0.025	0.113±0.019	0.112±0.014	0.374±0.018	0.744±0.009	0.097±0.007
PL-AGGD-PCA	0.649±0.019	0.627±0.017	0.540±0.019	0.081±0.012	0.085±0.018	0.305±0.015	0.444±0.007	0.077±0.007
PL-AGGD-DELIN	0.690±0.016	0.689±0.015	0.605±0.029	0.369±0.036	0.157±0.027	**0.465±0.016**	0.818±0.011	0.296±0.020
PL-AGGD-CENDA	0.697±0.022	**0.723±0.018**	0.604±0.028	0.377±0.043	0.190±0.018	0.394±0.022	0.850±0.007	0.301±0.016
PL-AGGD-DRAW	**0.702±0.024**	0.717±0.019	**0.615±0.022**	**0.409±0.027**	**0.209±0.050**	0.418±0.017	**0.894±0.009**	**0.304±0.013**

descending, i.e. the gradual weight descending is more consistent with the iteration process on candidate label disambiguation for partial label learning. When the value of s is equal to 1, the adaptive adjustment on weight between PCA and LDA is ablated, it is impressive that a obvious drop emerges in that situation. When the value of s is greater than 1, $s \in \{1.1, 1.2\}$, i.e. the weight of PCA increases, and the weight of LDA decreases, which is contrary to the behaviour as $s < 1$. Obviously, compared with reducing the weight of PCA weight, the classification accuracy drops with a great margin when $s > 1$, which validates that the increase of weight on supervised dimensionality reduction is reasonable for partial label learning paradigm.

For α, it is the initial value to balance the weight between the supervised dimensionality reduction and unsupervised dimensionality reduction. When $\alpha \simeq 1$, DRAW is dominated by PCA, and the weakly supervision from partial label instance is completely ignored. Accordingly, DRAW is dominated by LDA when $\alpha \simeq 0$. The projection matrix may be misled by the false positive labels of training instances without the guidance of unsupervised dimensionlity reduction. Therefore, we set the initial value of α to 0.5. As shown in Fig. 5, in terms of the choice of α, although classification accuracy on each partial label learning algorithm coupled with DRAW fluctuates with a certain magnitude with the change of α, the performance remains stable as long as the value of α is decreased during iterations. Furthermore, with the increase of s, the fluctuation on classification accuracy tends to be stable. In this paper, the value of k, α and s are set as 8, 0.5 and 0.95 respectively according to the comparative studies.

4.5 Further Analysis

Varying r. For partial label learning, models tend to fit false positive labels. Accordingly, on two representative datasets `Lost` and `tmc2007` ($r = 2$), we can observe typical overfitting phenomena as shown by the blue curves in Fig.8, where the gaps between training error and testing error are relatively large, when directly using IPAL and SURE. After employing DRAW, as shown by the yellow curves in Fig.8, the generalization gaps are greatly narrowed, and the testing classification error is obviously decreased. According to the results in Table 7, 8 and 9, it is easy to induce that the difficulty degree of partial label learning varies with the number of false positive labels. To demonstrate the relationship between the number of candidate labels and classification results, on datasets `mediamill`, `tmc2007`, `amazon` and `sport`, we control the number of false positive label in candidate label set (i.e., r) to increase from 1 to $q - 1$. Fig. 7 presents the classification accuracy of PL-KNN, IPAL coupled with PCA, DELIN, CENDA and the proposed DRAW. According to the results in Fig. 7, the following observations can be concluded:

- As the candidate label size increases, the supervision of training instances tends to be weakened, and

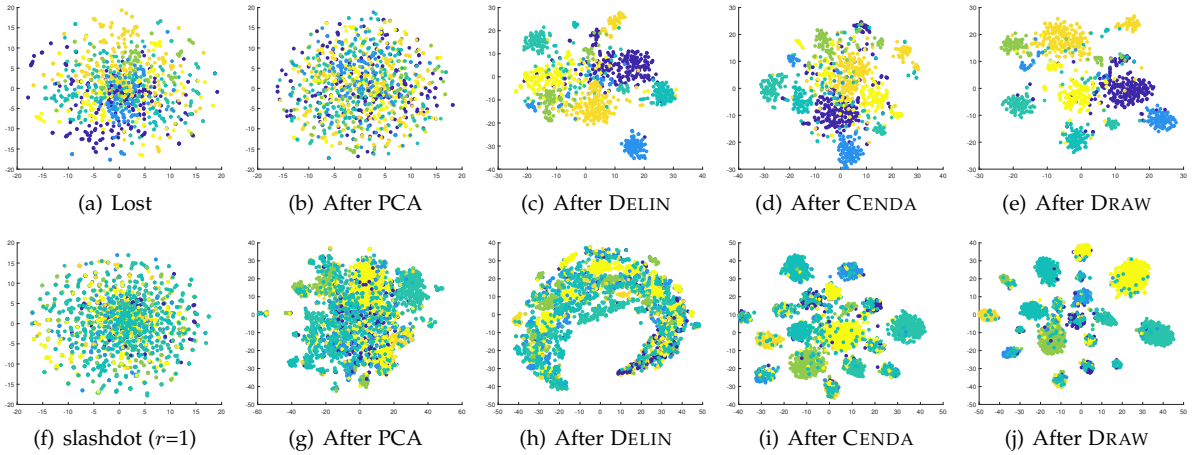
TABLE 10
Classification accuracy of $\mathcal{L}$-DRAW ($\mathcal{L} \in$ {PL-KNN, PL-SVM, IPAL, SURE, PL-AGGD}) changes with the number of retained features ($d' = \lceil ratio \cdot \min(q-1, d) \rceil$) varies with $ratio$ increases from 0.5 to 1 with an interval of 0.1. On each data set, the best performance across different values of $ratio$ is shown in bold face.

Data Set	$ratio$	# Retained Features	PL-KNN-DRAW	PL-SVM-DRAW	IPAL-DRAW	SURE-DRAW	PL-AGGD-DRAW
Lost	0.5	8	0.716±0.033	0.628±0.042	0.738±0.032	0.751±0.026	0.751±0.031
	0.6	9	0.735±0.027	0.658±0.044	0.771±0.040	0.780±0.034	0.776±0.036
	0.7	11	0.830±0.021	0.792±0.044	0.876±0.026	0.887±0.021	0.875±0.018
	0.8	12	0.833±0.019	0.787±0.057	0.877±0.042	0.884±0.019	0.876±0.020
	0.9	14	**0.842±0.018**	0.756±0.083	**0.890±0.028**	**0.889±0.024**	**0.886±0.026**
	1	15	0.832±0.018	**0.865±0.027**	0.888±0.030	0.885±0.024	0.883±0.027
FG-NET	0.5	39	0.153±0.028	0.102±0.016	0.144±0.032	0.139±0.027	0.139±0.027
	0.6	47	0.152±0.027	0.100±0.025	0.167±0.033	0.155±0.027	0.154±0.028
	0.7	54	0.158±0.035	0.082±0.014	**0.181±0.038**	0.150±0.030	0.146±0.032
	0.8	62	**0.164±0.038**	**0.105±0.024**	0.171±0.038	0.154±0.033	0.151±0.034
	0.9	70	0.149±0.038	0.085±0.022	0.178±0.046	**0.157±0.045**	**0.159±0.048**
	1	77	0.133±0.031	0.101±0.026	0.162±0.029	0.145±0.034	0.152±0.039
Mirflickr	0.5	7	**0.629±0.032**	0.531±0.112	**0.549±0.030**	0.683±0.021	0.654±0.028
	0.6	8	0.622±0.017	0.459±0.085	0.540±0.015	0.659±0.018	0.647±0.013
	0.7	10	0.618±0.027	0.520±0.100	0.545±0.021	0.659±0.024	0.651±0.021
	0.8	11	0.464±0.012	0.558±0.139	0.440±0.014	0.675±0.020	0.650±0.016
	0.9	12	0.499±0.021	**0.562±0.072**	0.451±0.024	**0.700±0.027**	**0.687±0.026**
	1	13	0.484±0.018	0.561±0.071	0.484±0.026	0.687±0.028	0.673±0.024
Yahoo! News	0.5	82	**0.543±0.008**	0.468±0.006	0.690±0.011	**0.657±0.008**	**0.669±0.009**
	0.6	98	0.523±0.008	0.466±0.008	**0.692±0.010**	0.654±0.011	0.667±0.011
	0.7	115	0.492±0.009	**0.471±0.005**	0.688±0.010	0.651±0.010	0.666±0.010
	0.8	131	0.472±0.008	0.464±0.008	0.684±0.009	0.646±0.009	0.662±0.012
	0.9	147	0.455±0.006	0.462±0.011	0.678±0.007	0.650±0.009	0.662±0.008
	1	163	0.443±0.006	0.465±0.010	0.675±0.007	0.645±0.009	0.663±0.008
Soccer Player	0.5	85	**0.501±0.003**	0.359±0.010	0.565±0.010	0.541±0.006	0.550±0.005
	0.6	102	0.498±0.002	0.373±0.012	0.566±0.010	0.543±0.006	0.551±0.005
	0.7	119	0.498±0.003	0.382±0.020	**0.567±0.010**	0.542±0.005	**0.554±0.005**
	0.8	136	0.497±0.003	0.379±0.012	0.563±0.009	**0.544±0.006**	0.549±0.004
	0.9	153	0.495±0.002	0.379±0.015	0.561±0.008	0.542±0.006	0.550±0.006
	1	170	0.495±0.003	**0.471±0.013**	0.562±0.008	0.541±0.006	0.550±0.007
BirdSong	0.5	6	0.619±0.024	0.508±0.038	0.625±0.021	0.627±0.018	0.637±0.017
	0.6	8	0.625±0.013	0.520±0.063	0.654±0.011	0.661±0.015	0.670±0.015
	0.7	9	0.637±0.019	0.526±0.033	0.686±0.011	0.683±0.014	0.687±0.016
	0.8	10	**0.643±0.013**	0.554±0.043	0.698±0.015	0.695±0.016	0.695±0.012
	0.9	11	0.642±0.012	**0.571±0.042**	0.700±0.012	0.720±0.024	0.709±0.016
	1	12	0.640±0.012	0.564±0.042	**0.706±0.014**	**0.724±0.022**	**0.712±0.012**
MSRCv2	0.5	11	**0.465±0.034**	0.368±0.059	0.478±0.035	0.474±0.021	0.480±0.023
	0.6	14	0.456±0.022	0.353±0.046	0.498±0.026	0.473±0.025	0.481±0.024
	0.7	16	0.448±0.024	0.351±0.050	0.509±0.030	0.477±0.017	0.487±0.031
	0.8	18	0.437±0.013	0.376±0.053	0.503±0.021	0.482±0.016	0.498±0.019
	0.9	20	0.457±0.019	**0.409±0.041**	0.522±0.033	0.488±0.024	0.503±0.024
	1	22	0.445±0.021	0.373±0.070	**0.540±0.016**	**0.490±0.023**	**0.505±0.030**

hence the classification performance of all methods decreases to some extents.

- Compared with PL-KNN and IPAL without dimensionality reduction, dimensionality reduction methods achieve better or comparable classification performance in most cases.
- Compared with PL-KNN, IPAL after using PCA, DELIN and CENDA, DRAW has consistently achieved superior classification performance in the majority of cases. Furthermore, although the improvements brought by dimensionality reduction is weakened with the increasing on the number of false positive labels, DRAW yields stable improvements against the others with varying r on these datasets.
- As shown in Fig. 7(h), when the rate of candidate labels in label space is greater than 94.7%, dimensionality reduction methods would not improve the generalization performance of IPAL on `sport`. According to these observations, it is easy to induce that the hard noisy instance with excessive number of candidate labels may mislead the supervised dimensionality reduction methods.

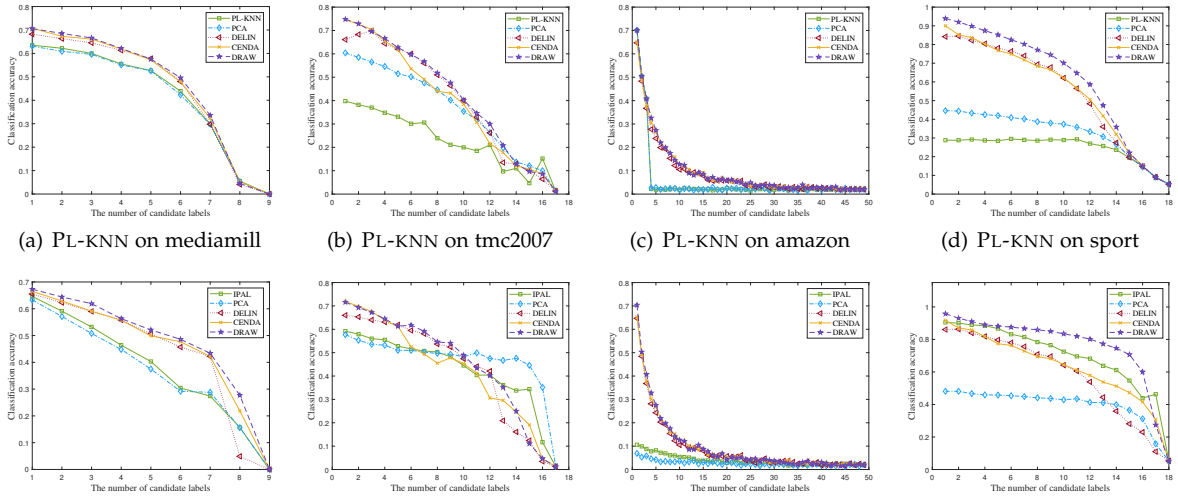
Experiments on image datasets. For the above-mentioned partial label datasets, the feature space is predefined by refering domain knowledge and heuristics. Thus, to further validate the effectiveness of proposed method, we leverage the neural network backbone as the feature extractor to conduct experiments on image datasets. Specifically, we use GoogleNet as the backbone model, which extracts 1024 dimensional features from the original features, on two image classification datasets `Oxford-Flowers-102` and `Caltech-101` datasets. We follow the same protocol as in the experiments on UCI datasets to generate six synthetic partial label datasets by adding 1/2/3



(a) Lost (b) After PCA (c) After DELIN (d) After CENDA (e) After DRAW

(f) slashdot (r=1) (g) After PCA (h) After DELIN (i) After CENDA (j) After DRAW

Fig. 6. Data visualization by t-SNE on real-world and synthetic partial label data sets before and after using dimensionality reduction methods.



(a) PL-KNN on mediamill (b) PL-KNN on tmc2007 (c) PL-KNN on amazon (d) PL-KNN on sport

(e) IPAL on mediamill (f) IPAL on tmc2007 (g) IPAL on amazon (h) IPAL on sport

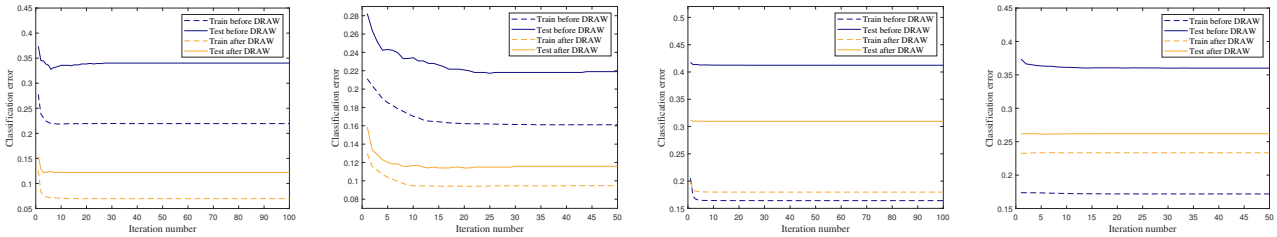
Fig. 7. Further analysis on the number of false positive label for PL-KNN and IPAL, which is coupled with four dimensionality reduction algorithms $\{$PCA, DELIN, CENDA, DRAW$\}$, classification accuracy changes as the number of false positive label in candidate label set (i.e., r) increases from 1 to $q-1$ with step size 1.

false positive labels to each sample in these datasets. The corresponding partial label datasets are denoted as `102flowers-f1`, `102flowers-f2`, `102flowers-f3`, `caltech101-f1`, `caltech101-f2` and `caltech101-f3`. As shown in Table 11, the experimental results show that DRAW is still helpful to improve the performance when combined with neural network-based feature extractor. Specifically, compared with the base partial label learning methods, the algorithms coupled with DRAW achieves superior performance in all cases. Compared with other dimensionality reduction methods, DRAW achieves superior performance against unsupervised dimensionality reduction method PCA, while achieves superior or at least comparable performance against DELIN and CENDA.

Visualization. In order to provide a more intuitive illustration of the effectiveness of DRAW, we employs the t-Distributed Stochastic Neighbor Embedding (t-SNE) algorithm to visualize the instances before and after using DRAW on real-world and synthetic partial label datasets. As shown in Fig. 6, compared with the instances in original feature space before using DRAW, the instances in lower-dimensional feature space after dimensionality reduction are generally more compact. We can observe that the specifically designed methods for partial label learning yield better separability compared with the unsupervised method PCA. Furthermore, DRAW achieves superior inter-class separability against PCA, DELIN and CENDA.

5 CONCLUSION

In this paper, we propose a novel approach for dimensionality reduction in the context of partial label learning. The proposed approach DRAW integrates both supervised dimensionality reduction LDA and unsupervised dimensionality reduction PCA via through adaptive weighting. The adaptive weight is determined by alternating between dimensionality reduction and candidate label set disambiguation. As the ground-truth label is gradually identified, the supervised dimensionality reduction becomes dominant in the induction of the projection matrix. Extensive comparative experiments conducted on synthetic and real-world partial label datasets demonstrate the effectiveness of DRAW



(a) IPAL on Lost (b) SURE on Lost (c) IPAL on tmc2007 (d) SURE on tmc2007

Fig. 8. Classification error on real-world and synthetic datasets for partial label learning algorithms IPAL and SURE. (a) Classification error of IPAL on real-world dataset `Lost` before and after using DRAW. (b) Classification error of SURE on real-world dataset `Lost` before and after using DRAW. (c) Classification error of IPAL on synthetic dataset `tmc2007` (r=2) before and using DRAW. (d) Classification error of SURE on synthetic dataset `tmc2007` (r=2) before and using DRAW.

TABLE 11
Classification accuracy (mean ± std) of each comparing algorithm on partial label data sets by using GoogleNet as feature extractor. For partial label learning algorithm $\mathcal{L} \in \{$PL-KNN, PL-SVM, IPAL, SURE, PL-AGGD$\}$, the performance of $\mathcal{L}$-DRAW ,$\mathcal{L}$-DELIN and $\mathcal{L}$-CENDA are compared against that of $\mathcal{L}$, where the best performance is shown in bold face.

Comparing Algorithm	Data Set					
	102flowers-f1	102flowers-f2	102flowers-f3	caltech101-f1	caltech101-f2	caltech101-f3
PL-KNN	0.710±0.012	0.703±0.012	0.696±0.013	0.864±0.010	0.861±0.010	0.861±0.010
PL-KNN-PCA	0.741±0.009	0.733±0.011	0.731±0.010	0.894±0.008	0.893±0.009	0.891±0.009
PL-KNN-DELIN	0.877±0.009	0.866±0.012	0.862±0.014	0.944±0.006	0.942±0.004	0.940±0.006
PL-KNN-CENDA	0.938±0.005	0.937±0.006	0.931±0.005	0.955±0.002	0.953±0.003	0.952±0.003
PL-KNN-DRAW	**0.940±0.005**	**0.937±0.005**	**0.936±0.005**	**0.970±0.005**	**0.970±0.004**	**0.968±0.003**
PL-SVM	0.808±0.009	0.796±0.018	0.790±0.014	0.879±0.015	0.876±0.037	0.873±0.023
PL-SVM-PCA	0.762±0.012	0.748±0.009	0.736±0.010	0.862±0.024	0.837±0.027	0.839±0.025
PL-SVM-DELIN	0.861±0.009	0.856±0.012	**0.843±0.011**	0.844±0.017	0.853±0.087	0.849±0.109
PL-SVM-CENDA	0.840±0.047	0.846±0.093	0.842±0.086	0.936±0.104	0.860±0.018	0.846±0.021
PL-SVM-DRAW	**0.895±0.008**	**0.872±0.022**	0.837±0.020	**0.945±0.028**	**0.911±0.147**	**0.905±0.044**
IPAL	0.831±0.010	0.829±0.010	0.827±0.009	0.917±0.008	0.916±0.008	0.914±0.007
IPAL-PCA	0.802±0.012	0.798±0.013	0.795±0.010	0.915±0.008	0.914±0.006	0.913±0.007
IPAL-DELIN	0.915±0.006	0.913±0.009	0.907±0.009	0.950±0.004	0.952±0.006	0.950±0.005
IPAL-CENDA	0.959±0.007	0.956±0.007	0.952±0.005	0.963±0.005	0.963±0.004	0.961±0.005
IPAL-DRAW	**0.965±0.004**	**0.961±0.004**	**0.957±0.005**	**0.964±0.006**	**0.966±0.004**	**0.963±0.005**
SURE	0.891±0.007	0.885±0.007	0.878±0.007	0.941±0.005	0.940±0.005	0.940±0.005
SURE-PCA	0.860±0.008	0.846±0.008	0.840±0.008	0.931±0.005	0.930±0.005	0.930±0.006
SURE-DELIN	0.924±0.005	0.917±0.011	0.911±0.008	0.955±0.004	0.954±0.005	0.953±0.006
SURE-CENDA	0.964±0.004	0.959±0.005	0.955±0.004	0.972±0.006	0.971±0.004	0.967±0.004
SURE-DRAW	**0.965±0.004**	**0.961±0.004**	**0.957±0.004**	**0.974±0.004**	**0.973±0.004**	**0.971±0.004**
PL-AGGD	0.889±0.006	0.881±0.008	0.869±0.006	0.940±0.005	0.939±0.004	0.936±0.004
PL-AGGD-PCA	0.860±0.009	0.849±0.009	0.838±0.008	0.931±0.006	0.931±0.005	0.928±0.005
PL-AGGD-DELIN	0.924±0.006	0.916±0.010	0.912±0.010	0.955±0.004	0.955±0.005	0.954±0.006
PL-AGGD-CENDA	0.964±0.004	0.959±0.005	0.954±0.005	0.972±0.006	0.972±0.004	0.967±0.004
PL-AGGD-DRAW	**0.965±0.004**	**0.961±0.004**	**0.957±0.005**	**0.975±0.004**	**0.973±0.004**	**0.971±0.004**

in significantly improving the generalization performance of well-established partial label learning algorithms. In future work, we aim to explore approaches that unify the dimensionality reduction and candidate label disambiguation into a single stage, further advancing the state-of-the-art in partial label learning.

REFERENCES

[1] C. Lee, S. Park, H. Song, J. Ryu, S. Kim, H. Kim, S. Pereira, and D. Yoo, "Interactive multi-class tiny-object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 14 136–14 145.

[2] Z.-H. Zhou, "Open-environment machine learning," *National Science Review*, vol. 9, no. 8, 2022.

[3] Y. Wang and J. Hu, "Global ridge orientation modeling for partial fingerprint identification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 33, no. 1, pp. 72–87, 2010.

[4] Z.-H. Zhou, "A brief introduction to weakly supervised learning," *National science review*, vol. 5, no. 1, pp. 44–53, 2018.

[5] D. Zhang, J. Han, G. Cheng, and M.-H. Yang, "Weakly supervised object localization and detection: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 44, no. 9, pp. 5866–5885, 2021.

[6] X. Qian, Y. Zeng, W. Wang, and Q. Zhang, "Co-saliency detection guided by group weakly supervised learning," *IEEE Transactions on Multimedia*, 2022.

[7] Y. Yan and Y. Guo, "Partial label learning with batch label correction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 04, 2020, pp. 6575–6582.

[8] N. Xu, C. Qiao, X. Geng, and M.-L. Zhang, "Instance-dependent partial label learning," *Advances in Neural Information Processing*

Systems, vol. 34, pp. 27 119–27 130, 2021.

[9] M. Guillaumin, J. Verbeek, and C. Schmid, "Multiple instance metric learning from automatically labeled bags of faces," in *Lecture Notes in Computer Science 6311*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin: Springer, 2010, pp. 634–647.

[10] Z. Zeng, S. Xiao, K. Jia, T.-H. Chan, S. Gao, D. Xu, and Y. Ma, "Learning by associating ambiguously labeled images," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 708–715.

[11] L. Liu and T. Dietterich, "A conditional multinomial mixture model for superset label learning," in *Advances in Neural Information Processing Systems 25*, P. Bartlett, F. C. N. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds. Cambridge, MA: MIT Press, 2012, pp. 557–565.

[12] W. Wang and M.-L. Zhang, "Partial label learning with discrimination augmentation," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 1920–1928.

[13] J. Lv, M. Xu, L. Feng, G. Niu, X. Geng, and M. Sugiyama, "Progressive identification of true labels for partial-label learning," in *International Conference on Machine Learning*. PMLR, 2020, pp. 6500–6510.

[14] J. Fan, Y. Yu, Z. Wang, and J. Gu, "Partial label learning based on disambiguation correction net with graph representation," *IEEE Transactions on Circuits and Systems for Video Technology*, 2021.

[15] F. Briggs, X. Z. Fern, and R. Raich, "Rank-loss support instance machines for MIML instance annotation," in *Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Beijing, China, 2012, pp. 534–542.

[16] D.-B. Wang, M.-L. Zhang, and L. Li, "Adaptive graph guided disambiguation for partial label learning," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 01, pp. 1–1, 2021.

[17] D. Zhou, Z. Zhang, M.-L. Zhang, and Y. He, "Weakly supervised POS tagging without disambiguation," *ACM Transactions on Asian and Low-Resource Language Information Processing*, vol. 17, no. 4, p. Article 35, 2018.

[18] M.-L. Zhang, J.-H. Wu, and W.-X. Bao, "Disambiguation enabled linear discriminant analysis for partial label dimensionality reduction," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 16, no. 4, pp. 1–18, 2022.

[19] W.-X. Bao, J.-Y. Hang, and M.-L. Zhang, "Partial label dimensionality reduction via confidence-based dependence maximization," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 46–54.

[20] A. J. Izenman, "Linear discriminant analysis," in *Modern multivariate statistical techniques*. Springer, 2013, pp. 237–280.

[21] F. Kherif and A. Latypova, "Principal component analysis," in *Machine Learning*. Elsevier, 2020, pp. 209–225.

[22] M.-K. Xie and S.-J. Huang, "Partial multi-label learning with noisy label identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[23] Y. Tian, X. Yu, and S. Fu, "Partial label learning: Taxonomy, analysis and outlook," *Neural Networks*, 2023.

[24] M.-L. Zhang and Z.-H. Zhou, "A review on multi-label learning algorithms," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 8, pp. 1819–1837, 2013.

[25] Z. Song, X. Yang, Z. Xu, and I. King, "Graph-based semi-supervised learning: A comprehensive review," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.

[26] W. Chen, G. Li, X. Zhang, S. Wang, L. Li, and Q. Huang, "Weakly supervised text-based actor-action video segmentation by clip-level multi-instance learning," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 2022.

[27] T. Yuan, F. Wan, M. Fu, J. Liu, S. Xu, X. Ji, and Q. Ye, "Multiple instance active learning for object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 5330–5339.

[28] S.-J. Huang, W. Gao, and Z.-H. Zhou, "Fast multi-instance multi-label learning," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 11, pp. 2614–2627, 2018.

[29] Y. Ma, C. Cui, X. Nie, G. Yang, K. Shaheed, and Y. Yin, "Pre-course student performance prediction with multi-instance multi-label learning," *Science China Information Sciences*, vol. 62, no. 2, pp. 1–3, 2019.

[30] W. Ji and R. Wang, "A multi-instance multi-label dual learning approach for video captioning," *ACM Transactions on Multimidia Computing Communications and Applications*, vol. 17, no. 2s, pp. 1–18, 2021.

[31] M.-L. Zhang, B.-B. Zhou, and X.-Y. Liu, "Partial label learning via feature-aware disambiguation," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, 2016, pp. 1335–1344.

[32] M.-K. Xie, F. Sun, and S.-J. Huang, "Partial multi-label learning with meta disambiguation," in *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 2021, pp. 1904–1912.

[33] M.-L. Zhang and F. Yu, "Solving the partial label learning problem: An instance-based approach," in *Proceedings of the 24th International Joint Conference on Artificial Intelligence*, Buenos Aires, Argentina, 2015, pp. 4048–4054.

[34] T. Cour, B. Sapp, and B. Taskar, "Learning from partial labels," *Journal of Machine Learning Research*, vol. 12, no. May, pp. 1501–1536, 2011.

[35] R. Jin and Z. Ghahramani, "Learning with multiple labels," in *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer, Eds. Cambridge, MA: MIT Press, 2003, pp. 897–904.

[36] F. Yu and M.-L. Zhang, "Maximum margin partial label learning," *Machine Learning*, vol. 106, no. 4, pp. 573–593, 2017.

[37] C.-Z. Tang and M.-L. Zhang, "Confidence-rated discriminative partial label learning," in *Proceedings of the 31st AAAI Conference on Artificial Intelligence*, San Francisco, CA, 2017, pp. 2611–2617.

[38] T. Cour, B. Sapp, and B. Taskar, "Learning from partial labels," *The Journal of Machine Learning Research*, vol. 12, pp. 1501–1536, 2011.

[39] M.-L. Zhang, F. Yu, and C.-Z. Tang, "Disambiguation-free partial label learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 10, pp. 2155–2167, 2017.

[40] X. Wu and M.-L. Zhang, "Towards enabling binary decomposition for partial label learning." in *IJCAI*, 2018, pp. 2868–2874.

[41] M. Wang, H. Han, Z. Huang, and J. Xie, "Unsupervised spectral feature selection algorithms for high dimensional data," *Frontiers of Computer Science*, vol. 17, no. 5, p. 175330, 2023.

[42] C. Tang, X. Zheng, W. Zhang, X. Liu, X. Zhu, and E. Zhu, "Unsupervised feature selection via multiple graph fusion and feature weight learning," *Science China Information Sciences*, vol. 66, no. 5, pp. 1–17, 2023.

[43] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

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

[45] M. Belkin and P. Niyogi, "Convergence of laplacian eigenmaps," *Advances in neural information processing systems*, vol. 19, 2006.

[46] X. He and P. Niyogi, "Locality preserving projections," *Advances in neural information processing systems*, vol. 16, 2003.

[47] J. Bourgain, "On lipschitz embedding of finite metric spaces in hilbert space," *Israel Journal of Mathematics*, vol. 52, pp. 46–52, 1985.

[48] W. B. Johnson, "Extensions of lipshitz mapping into hilbert space," in *Conference modern analysis and probability, 1984*, 1984, pp. 189–206.

[49] Y. Zeng, Y. Tong, and L. Chen, "Hst+: An efficient index for embedding arbitrary metric spaces," in *2021 IEEE 37th International Conference on Data Engineering (ICDE)*. IEEE, 2021, pp. 648–659.

[50] Y. Zeng, Y. Tong, and L. Chen, "Faster and better solution to embed lp metrics by tree metrics," in *Proceedings of the 2022 International Conference on Management of Data*, 2022, pp. 2135–2148.

[51] Y. Zeng, Y. Tong, and L. Chen, "Litehst: A tree embedding based method for similarity search," *Proceedings of the ACM on Management of Data*, vol. 1, no. 1, pp. 1–26, 2023.

[52] K.-K. Wang, G.-P. Yang, L. Yang, Y.-W. Huang, and Y.-L. Yin, "Ecg biometrics via enhanced correlation and semantic-rich embedding," *Machine Intelligence Research*, pp. 1–10, 2023.

[53] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Philosophical transactions of the royal society A: Mathematical, Physical and Engineering Sciences*, vol. 374, no. 2065, 2016.

[54] I. Jolliffe, "A 50-year personal journey through time with principal component analysis," *Journal of Multivariate Analysis*, vol. 188, p. 104820, 2022.

[55] C. G. M. Snoek, M. Worring, J. C. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders, "The challenge problem for automated detection of 101 semantic concepts in multimedia," in *Proceedings*

of the 14th ACM International Conference on Multimedia, Santa Barbara, CA, 2006, pp. 421–430.
[56] A. N. Srivastava and B. Zane-Ulman, "Discovering recurring anomalies in text reports regarding complex space systems," in *Proceedings of the 2005 IEEE Aerospace Conference*, Big Sky, MT, 2005.
[57] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney, "Community structure in large networks: Natural cluster sizes and the absence of large well-defined clusters," *Internet Mathematics*, vol. 6, no. 1, pp. 29–123, 2009.
[58] D. Dheeru and E. K. Taniskidou, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml
[59] H. Soleimani and D. J. Miller, "Semi-supervised multi-label topic models for document classification and sentence labeling," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, Indianapolis, IN, 2016, pp. 105–114.
[60] I. Katakis, G. Tsoumakas, and I. Vlahavas, "Multilabel text classification for automated tag suggestion," in *Proceedings of the ECML/PKDD 2008 Discovery Challenge*, Antwerp, Belgium, 2008.
[61] K. Altun and B. Barshan, "Human activity recognition using inertial/magnetic sensor units," in *Proceedings of the 1st International Conference on Human Behavior Understanding*, Istanbul, Turkey, 2010, pp. 38–51.
[62] J. D. M. Rennie and R. Rifkin, "Improving multiclass text classification with the support vector machines," Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Tech. Rep. AIM-2001-026, 2001.
[63] G. Panis and A. Lanitis, "An overview of research activities in facial age estimation using the fg-net aging database," in *Lecture Notes in Computer Science 8926*, C. R. L. Agapito, M. M. Bronstein, Ed. Berlin: Springer, 2015, pp. 737–750.
[64] M. J. Huiskes and M. S. Lew, "The mir flickr retrieval evaluation," in *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*, Vancouver, Canada, 2008, pp. 39–43.
[65] Z. Zeng, S. Xiao, K. Jia, T.-H. Chan, S. Gao, D. Xu, and Y. Ma, "Learning by associating ambiguously labeled images," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Portland, OR, 2013, pp. 708–715.
[66] E. Hüllermeier and J. Beringer, "Learning from ambiguously labeled examples," *Intelligent Data Analysis*, vol. 10, no. 5, pp. 419–439, 2006.
[67] N. Nguyen and R. Caruana, "Classification with partial labels," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Las Vegas, NV, 2008, pp. 381–389.
[68] L. Feng and B. An, "Partial label learning with self-guided retraining," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, 2019, pp. 3542–3549.
[69] H. Wang, R. Xiao, Y. Li, L. Feng, G. Niu, G. Chen, and J. Zhao, "Pico: Contrastive label disambiguation for partial label learning," in *International Conference on Learning Representations*, 2021.
[70] L. Feng, J. Lv, B. Han, M. Xu, G. Niu, X. Geng, B. An, and M. Sugiyama, "Provably consistent partial-label learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 10 948–10 960, 2020.