Disambiguation Enabled Linear Discriminant Analysis for Partial Label Dimensionality Reduction

MIN-LING ZHANG, JING-HAN WU, WEI-XUAN BAO, Southeast University, China

As an emerging weakly supervised learning framework, partial label learning considers inaccurate supervision where each training example is associated with multiple *candidate* labels among which only one is valid. In this paper, a first attempt towards employing dimensionality reduction to help improve the generalization performance of partial label learning system is investigated. Specifically, the popular linear discriminant analysis (LDA) techniques are endowed with the ability of dealing with partial label training examples. To tackle the challenge of unknown ground-truth labeling information, a novel learning approach named Delin is proposed which alternates between LDA dimensionality reduction and candidate label disambiguation based on estimated labeling confidences over candidate labels. On one hand, the (kernelized) projection matrix of LDA is optimized by utilizing disambiguation-guided labeling confidences. On the other hand, the labeling confidences are disambiguated by resorting to kNN aggregation in the LDA-induced feature space. Extensive experiments over a broad range of partial label data sets clearly validate the effectiveness of Delin in improving the generalization performance of well-established partial label learning algorithms.

CCS Concepts: • **Computing methodologies → Machine learning algorithms**; *Learning paradigms*;

Additional Key Words and Phrases: partial label learning, dimensionality reduction, linear discriminant analysis, disambiguation

ACM Reference format:
Min-Ling Zhang, Jing-Han Wu, Wei-Xuan Bao. 2021. Disambiguation Enabled Linear Discriminant Analysis for Partial Label Dimensionality Reduction. *ACM Trans. Knowl. Discov. Data.* XX, XX, Article XXX (XX 2021), 18 pages.
https://doi.org/XXXXX

1 INTRODUCTION

In partial label (PL) learning [11, 28, 53], each training example is represented by a single instance while associated with a set of *candidate* labels among which only one corresponds to the ground-truth label. The task of partial label learning is to learn a multi-class classification model from PL training examples which is capable of assigning proper class label for the unseen instance. As an emerging weakly supervised learning framework with inaccurate supervision [57], the need of learning from examples with candidate label sets naturally arises under many real-world scenarios, such as web mining [21], multimedia content analysis [8, 10, 26, 51], ecoinformatics [3, 26, 42], natural language processing [33, 34, 56], etc.

Due to the limited supervision information available from training set, the generalization performance of partial label learning system is usually less satisfactory. As an effective way to help improve the generalization ability of learning system, it is rather desirable to explore beneficial dimensionality reduction mechanism for partial label learning.

Manuscript submitted to ACM

Existing works mainly focus on inducing partial label classification model by disambiguating the candidate label sets of PL training examples [6, 8, 9, 11, 17, 26, 27, 42, 50], while the usefulness of dimensionality reduction for partial label learning hasn't been well investigated. Accordingly, in order to design partial label dimensionality reduction techniques, the major challenge lies in that the ground-truth label of each PL training example is not directly accessible to the learning algorithm.

To tackle the challenge of unknown ground-truth labeling information, a first attempt towards partial label dimensionality reduction is investigated in this paper where a novel learning approach named Delin, i.e. *Disambiguation Enabled LINear discriminant analysis*, is proposed. Specifically, Delin endows the popular linear discriminant analysis (LDA) techniques with the ability of dealing with PL training examples. Based on labeling confidences estimated over candidate labels, an alternating procedure is employed by Delin to enable LDA dimensionality reduction and candidate label disambiguation. On one hand, LDA dimensionality reduction is performed by optimizing the (kernelized) projection matrix via the utilization of disambiguation-guided labeling confidences. On the other hand, candidate label disambiguation is performed by resorting to *k*NN aggregation in the feature space induced by LDA projection matrix. Comprehensive experiments over synthetic and real-world partial label data sets show that Delin serves as an effective dimensionality reduction approach to improving the generalization performance of well-established partial label learning algorithms.

The rest of this paper is organized as follows. Section 2 briefly discusses related works on partial label learning. Section 3 introduces technical procedure of the proposed Delin approach. Section 4 reports detailed results of experimental studies. Finally, Section 5 concludes this paper.

2 RELATED WORKS

Partial label learning is one of the emerging weakly supervised learning frameworks [57], where the learning system needs to learn from *inaccurate* supervision information with the ground-truth label concealed in the candidate label set of each training example. Conceptually speaking, partial label learning is related to several well-established weakly supervised learning frameworks such as *semi-supervised learning*, *multi-instance learning* and *multi-label learning*.

Semi-supervised learning works under *incomplete* supervision where only few labeled examples are available for training along with abundant unlabeled examples [7, 44, 59]. Although the ground-truth label for either unlabeled example or PL example is unknown to the learning algorithm, the scope of ground-truth label for unlabeled example and PL example assumes the whole label space and the candidate label set respectively. Multi-instance learning works under *inexact* supervision where the class label is assigned at the level of bags (i.e. a set of instances) instead of individual instances [2, 5, 25]. Although the actual correspondence between instances and labels for either multi-instance example or PL example is ambiguous, the ambiguity for multi-instance example and PL example arises in the instance space and the label space respectively. Multi-label learning works under *non-unique* supervision where multiple valid class labels are assigned to a single instance [47, 54, 58]. Although the labeling information for either multi-label example or PL example is non-unique, the set of class labels assigned to multi-instance example and PL example are valid and candidate ones respectively.

To learn from PL training examples, one natural solution is trying to recover the ground-truth labeling information via candidate label set disambiguation, including *disambiguation by identification* or *disambiguation by averaging*. For the strategy of identification-based disambiguation, the unknown ground-truth label is treated as latent variable whose value is estimated by employing iterative optimization procedure such as EM. The optimization objective can be instantiated in different ways such as maximizing the likelihood of observing the PL training examples over their candidate label sets [22, 26, 27], or maximizing the predictive margin between candidate labels and non-candidate labels of PL training examples [6, 15, 31, 43, 50].

For the strategy of averaging-based disambiguation, all candidate labels of the PL training example are treated in an equal manner whose modeling outputs are averaged to help yield the final prediction. The averaging procedure can be instantiated in different ways such as distinguishing the averaged modeling outputs from candidate labels between the modeling outputs from non-candidate labels for discriminative models [11, 41], or aggregating the votes among candidate labels of the unseen instance's neighboring examples for distance-based models [17, 20, 39, 52].

The effectiveness of disambiguation strategy could be impacted by the false positive labels residing in the candidate label set, especially when the size of candidate label set is large. Specifically, for identification-based disambiguation the estimated ground-truth label might turn out to be false positive one, while for averaging-based disambiguation the modeling output from false positive labels might overwhelm the intrinsic modeling output from ground-truth label. Other than candidate label set disambiguation, another possible solution is trying to transform the problem of learning from PL examples into other learning problems in principled ways. Correspondingly, existing instantiations towards the transformation strategy can be witnesses such as binary decomposition [46, 53], dictionary learning [9], graph matching [29], regression [14, 42, 48, 55], online learning [49], etc.

It is worth noting that existing works on partial label learning mainly focus on manipulating the label space to induce the PL predictive model, while few attempts have been made in manipulating the instance space to help improve the generalization ability of partial label learning system. Next, we present the Delin approach which makes use of the popular linear discriminant analysis techniques to learn from PL training examples.

3 THE PROPOSED APPROACH

Let $\mathcal{X} = \mathbb{R}^d$ be the d-dimensional instance space and $\mathcal{Y} = \{l_1, l_2, \ldots, l_q\}$ be the label space consisting of q class labels. Given the PL training set $\mathcal{D} = \{(\boldsymbol{x}_i, S_i) \mid 1 \leq i \leq m\}$ where $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{id})^\top \in \mathcal{X}$ is a d-dimensional feature vector and $S_i \subseteq \mathcal{Y}$ is the candidate label set associated with $\boldsymbol{x}_i$, the task of partial label learning is to derive a *multi-class* classification model $f : \mathcal{X} \mapsto \mathcal{Y}$ from $\mathcal{D}$. For each PL training example $(\boldsymbol{x}_i, S_i)$, it is assumed that the ground-truth label y_i for $\boldsymbol{x}_i$ resides in its candidate label set S_i (i.e. $y_i \in S_i$) while is not directly accessible to the training algorithm.

Let $\mathbf{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m] \in \mathbb{R}^{d \times m}$ be the instance matrix formed by concatenating all feature vectors in the training set, the goal of partial label dimensionality reduction is to learn a projection matrix $\mathbf{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'}] \in \mathbb{R}^{d \times d'}$ $(d' \ll d)$ which maps $\mathbf{X}$ into the d'-dimensional feature space, i.e. $\mathbf{X}' = \mathbf{W}^\top \mathbf{X}$. In this paper, we propose the Delin approach by adapting the popular linear discriminant analysis mechanism, where the projection matrix $\mathbf{W}$ is learned iteratively via an alternating procedure between *LDA dimensionality reduction* and *candidate label disambiguation.*

To fulfill the alternating procedure, we make use of the labeling confidence matrix $\mathbf{Y} = [Y_{ij}]_{m \times q}$ where each element Y_{ij} represents the estimated confidence of l_j being the ground-truth label for $\boldsymbol{x}_i$. Specifically, the labeling confidence matrix is initialized as follows:

$$\forall\ 1 \leq i \leq m,\ 1 \leq j \leq q :\ Y_{ij} = \begin{cases} \frac{1}{|S_i|}, & \text{if } l_j \in S_i \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Here, the constraints $\sum_{j=1}^{q} Y_{ij} = 1$ $(1 \leq i \leq m)$ hold in accordance with the PL assumption that the ground-truth label of $\boldsymbol{x}_i$ resides in its candidate label set S_i.

In the following subsections, technical details of the two alternating steps are scrutinized.

3.1 LDA Dimensionality Reduction

For traditional multi-class LDA [16, 30], the projection matrix $\mathbf{W}$ is learned by solving the following optimization problem:

$$\begin{aligned} &\arg\max_{\mathbf{W}} \ \operatorname{tr}\left(\mathbf{W}^{\top}\mathrm{S}_b\mathbf{W}\right) \\ &\text{s.t.}: \ \boldsymbol{w}_h^{\top}\mathrm{S}_w\boldsymbol{w}_h = 1 \quad (1 \leq h \leq d') \end{aligned} \tag{2}$$

Here, $\mathrm{S}_b \in \mathbb{R}^{d\times d}$ and $\mathrm{S}_w \in \mathbb{R}^{d\times d}$ correspond to the *between-class* scatter matrix and *within-class* scatter matrix respectively.

To endow LDA with the ability of dealing with PL training examples, the key adaptation lies in the derivation of scatter matrices S_b and S_w. Let $\mathbf{Y}$ be the current labeling confidence matrix, Delin specifies the global mean vector $\boldsymbol{\mu} \in \mathbb{R}^d$ and the class-wise mean vector $\boldsymbol{\mu}_j \in \mathbb{R}^d$ $(1 \leq j \leq q)$ as follows:

$$\begin{aligned} \boldsymbol{\mu} &= \frac{\sum_{i=1}^{m}\boldsymbol{x}_i}{m} \\ \boldsymbol{\mu}_j &= \frac{\sum_{i=1}^{m} Y_{ij}\cdot\boldsymbol{x}_i}{\sum_{i=1}^{m} Y_{ij}} \end{aligned} \tag{3}$$

Accordingly, the total scatter matrix $\mathrm{S}_t \in \mathbb{R}^{d\times d}$ and within-class scatter matrix S_w can be derived as:

$$\begin{aligned} \mathrm{S}_t &= \sum_{i=1}^{m}(\boldsymbol{x}_i-\boldsymbol{\mu})(\boldsymbol{x}_i-\boldsymbol{\mu})^{\top} \\ &= \bar{\mathbf{X}}^{\top}\bar{\mathbf{X}} \\ \mathrm{S}_w &= \sum_{j=1}^{q}\sum_{i=1}^{m} Y_{ij}\cdot(\boldsymbol{x}_i-\boldsymbol{\mu}_j)(\boldsymbol{x}_i-\boldsymbol{\mu}_j)^{\top} \end{aligned} \tag{4}$$

Here, $\bar{\mathbf{X}} = \mathbf{X} - \boldsymbol{\mu}\boldsymbol{e}^{\top}$ represents the centralized instance matrix with $\boldsymbol{e} = [1, 1, \ldots, 1]^{\top}$ being an m-dimensional unit vector. Thereafter, the between-class scatter matrix S_b can be derived as:

$$\begin{aligned} \mathrm{S}_b &= \mathrm{S}_t - \mathrm{S}_w \\ &= \sum_{j=1}^{q}\left(\sum_{i=1}^{m} Y_{ij}\right)\cdot(\boldsymbol{\mu}_j-\boldsymbol{\mu})(\boldsymbol{\mu}_j-\boldsymbol{\mu})^{\top} \\ &= \bar{\mathbf{X}}^{\top}\mathbf{Y}\mathbf{C}^{-1}\mathbf{Y}^{\top}\bar{\mathbf{X}} \end{aligned} \tag{5}$$

Here, $\mathbf{C} = \operatorname{diag}[c_1, c_2, \ldots, c_q]$ represents the $q \times q$ diagonal matrix with diagonal element $c_j = \sum_{i=1}^{m} Y_{ij}$ $(1 \leq j \leq q)$.

For each projection vector $\boldsymbol{w}_h$ $(1 \leq h \leq d')$ in $\mathbf{W}$, the Lagrange function w.r.t. Eq.(2) can be derived by introducing Lagrange multipliers λ_h:

$$L(\boldsymbol{w}_h, \lambda_h) = \boldsymbol{w}_h^{\top}\mathrm{S}_b\boldsymbol{w}_h - \lambda_h(\boldsymbol{w}_h^{\top}\mathrm{S}_w\boldsymbol{w}_h - 1) \tag{6}$$

By setting $\frac{\partial L(\boldsymbol{w}_h, \lambda_h)}{\partial \boldsymbol{w}_h} = \mathbf{0}$, the necessary condition on the optimal solution of $\boldsymbol{w}_h$ corresponds to:

$$\left(\mathrm{S}_w^{-1}\mathrm{S}_b\right)\boldsymbol{w}_h = \lambda_h \boldsymbol{w}_h \tag{7}$$

Therefore, λ_h and $\boldsymbol{w}_h$ turn out to be an eigenvalue and the corresponding eigenvector of $\mathrm{S}_w^{-1}\mathrm{S}_b$. Accordingly, Delin forms the LDA projection matrix $\mathbf{W}$ by concatenating the eigenvectors w.r.t. the top d' eigenvalues of $\mathrm{S}_w^{-1}\mathrm{S}_b$.

Kernel Extension Other than the above plain procedure, kernel trick can also be introduced to perform LDA dimensionality reduction. Let $\varphi : \mathbb{R}^d \mapsto \mathbb{R}^{\mathcal{H}_\kappa}$ be the (implicit) mapping from the original feature space to the Reproducing Kernel Hilbert Space (RKHS) induced by kernel function $\kappa : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R}$. Accordingly, we can have the global mean vector $\boldsymbol{\mu}^\kappa \in \mathbb{R}^{\mathcal{H}_\kappa}$ and the class-wise mean vector $\boldsymbol{\mu}_j^\kappa \in \mathbb{R}^{\mathcal{H}_\kappa}$ $(1 \leq j \leq q)$ in RKHS:

$$\boldsymbol{\mu}^\kappa = \frac{\sum_{i=1}^m \varphi(\boldsymbol{x}_i)}{m} \tag{8}$$

$$\boldsymbol{\mu}_j^\kappa = \frac{\sum_{i=1}^m Y_{ij} \cdot \varphi(\boldsymbol{x}_i)}{\sum_{i=1}^m Y_{ij}} \tag{9}$$

Therefore, the kernelized between-class scatter matrix S_b^κ and within-class scatter matrix S_w^κ can be derived as follows:

$$\mathrm{S}_b^\kappa = \sum_{j=1}^q \left(\boldsymbol{\mu}_j^\kappa - \boldsymbol{\mu}^\kappa\right)\left(\boldsymbol{\mu}_j^\kappa - \boldsymbol{\mu}^\kappa\right)^\top \tag{10}$$

$$\mathrm{S}_w^\kappa = \sum_{j=1}^q \sum_{i=1}^m Y_{ij} \cdot \left(\varphi(\boldsymbol{x}_i) - \boldsymbol{\mu}_j^\kappa\right)\left(\varphi(\boldsymbol{x}_i) - \boldsymbol{\mu}_j^\kappa\right)^\top \tag{11}$$

Without loss of generality, let $\boldsymbol{w}_h^\kappa$ $(1 \leq h \leq d')$ be the projection vector in RKHS:

$$\boldsymbol{w}_h^\kappa = \varphi(\mathrm{X})\boldsymbol{\alpha}_h \tag{12}$$

where $\varphi(\mathrm{X}) = [\varphi(\boldsymbol{x}_1), \varphi(\boldsymbol{x}_2), \ldots, \varphi(\boldsymbol{x}_m)]$ corresponds to the instance matrix in RKHS and $\boldsymbol{\alpha}_h = [\alpha_1^h, \alpha_2^h, \ldots, \alpha_m^h]^\top$ corresponds to the coefficient vector to be learned for kernelized LDA.

Then, the between-class variance w.r.t. projection vector $\boldsymbol{w}_h^\kappa$ in RKHS can be calculated as:

$$\begin{aligned} g_b(\boldsymbol{\alpha}_h) &= \boldsymbol{w}_h^{\kappa\top}\,\mathrm{S}_b^\kappa\,\boldsymbol{w}_h^\kappa \\ &= \boldsymbol{\alpha}_h^\top \varphi(\mathrm{X})^\top \left(\sum_{j=1}^q \left(\boldsymbol{\mu}_j^\kappa - \boldsymbol{\mu}^\kappa\right)\left(\boldsymbol{\mu}_j^\kappa - \boldsymbol{\mu}^\kappa\right)^\top\right)\varphi(\mathrm{X})\boldsymbol{\alpha}_h \\ &= \boldsymbol{\alpha}_h^\top\,\mathbf{\Phi}\,\boldsymbol{\alpha}_h \end{aligned} \tag{13}$$

Here, $\mathbf{\Phi} = \sum_{j=1}^q (\boldsymbol{\phi}_j - \boldsymbol{\phi}^*)(\boldsymbol{\phi}_j - \boldsymbol{\phi}^*)^\top \in \mathbb{R}^{m \times m}$ with $\boldsymbol{\phi}^* = [\phi_1^*, \phi_2^*, \ldots, \phi_m^*]^\top$ and $\boldsymbol{\phi}_j = [\phi_{j1}, \phi_{j2}, \ldots, \phi_{jm}]^\top$ $(1 \leq j \leq q)$ taking the following component values:

$$\begin{aligned} \phi_i^* &= \frac{\sum_{k=1}^m \kappa(\boldsymbol{x}_i, \boldsymbol{x}_k)}{m} \;(1 \leq i \leq m) \\ \phi_{ji} &= \frac{\sum_{k=1}^m Y_{kj} \cdot \kappa(\boldsymbol{x}_i, \boldsymbol{x}_k)}{\sum_{k=1}^m Y_{kj}} \;(1 \leq i \leq m) \end{aligned} \tag{14}$$

Accordingly, the within-class variance w.r.t. projection vector $\boldsymbol{w}_h^\kappa$ in RKHS can be calculated as:

$$\begin{aligned} g_w(\boldsymbol{\alpha}_h) &= \boldsymbol{w}_h^{\kappa\top}\, \mathrm{S}_w^{\kappa}\, \boldsymbol{w}_h^{\kappa} \\ &= \boldsymbol{w}_h^{\kappa\top} \left(\sum_{j=1}^{q} \sum_{i=1}^{m} Y_{ij} \cdot \left(\varphi(\boldsymbol{x}_i) - \boldsymbol{\mu}_j^{\kappa} \right) \left(\varphi(\boldsymbol{x}_i) - \boldsymbol{\mu}_j^{\kappa} \right)^{\top} \right) \boldsymbol{w}_h^{\kappa} \\ &= \boldsymbol{\alpha}_h^{\top}\, \boldsymbol{\Omega}\, \boldsymbol{\alpha}_h \end{aligned} \tag{15}$$

Here, $\boldsymbol{\Omega} = \sum_{j=1}^{q} \sum_{i=1}^{m} Y_{ij} \cdot \mathbf{K}\mathbf{H}_j\mathbf{K}^{\top} \in \mathbb{R}^{m\times m}$ with $\mathbf{K} = \varphi(\mathbf{X})^{\top}\varphi(\mathbf{X}) = [K_{ij}]_{m\times m}$ being the Gram matrix, i.e. $K_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$. Furthermore, $\mathbf{H}_j \in \mathbb{R}^{m\times m}$ turns out to be the following matrix:

$$\mathbf{H}_j = \mathbf{I} - \frac{\mathbf{1}\mathbf{1}^{\top}}{\sum_{i=1}^{m} Y_{ij}} \;(1 \le j \le q) \tag{16}$$

Here, $\mathbf{I}$ and $\mathbf{1}$ represent the $m \times m$ identity matrix and m-dimensional all-one vector respectively.

Similar to Eq.(2), the coefficient matrix $\mathbf{A} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_{d'}] \in \mathbb{R}^{m\times d'}$ for kernelized LDA can be learned by solving the following problem:

$$\begin{aligned} &\arg\max_{\mathbf{A}} \;\mathrm{tr}\left(\mathbf{A}^{\top}\boldsymbol{\Phi}\mathbf{A}\right) \\ &\text{s.t. : } \boldsymbol{\alpha}_h^{\top}\, \boldsymbol{\Omega}\, \boldsymbol{\alpha}_h = 1 \quad (1 \le h \le d') \end{aligned} \tag{17}$$

In the kernelized version, Delin forms the coefficient matrix $\mathbf{A}$ by concatenating the eigenvectors w.r.t. the top d' eigenvalues of $\boldsymbol{\Omega}^{-1}\boldsymbol{\Phi}$. Accordingly, the instance matrix $\mathbf{X}$ can be mapped into the d'-dimensional feature space, i.e. $\mathbf{X}' = \mathbf{A}^{\top}\mathbf{K}$.

3.2 Candidate Label Disambiguation

Given the mapped instance matrix $\mathbf{X}' = [\boldsymbol{x}_1', \boldsymbol{x}_2', \ldots, \boldsymbol{x}_m'] \in \mathbb{R}^{d'\times m}$, we can have the transformed PL training set in LDA-induced feature space, i.e. $\mathcal{D}' = \{(\boldsymbol{x}_i', S_i) \mid 1 \le i \le m\}$. Then, the labeling confidence matrix $\mathbf{Y}$ will be updated to $\mathbf{Y}' = [Y_{ij}']_{m\times q}$ by exploiting the transformed PL training examples.

Specifically, Delin performs kNN aggregation in the LDA-induced feature space. For each instance $\boldsymbol{x}_i' \in \mathbb{R}^{d'}$, we use $\mathcal{N}(\boldsymbol{x}_i')$ to denote its k nearest neighbors identified in $\mathcal{D}'$. Afterwards, a counting matrix $\mathbf{V} = [V_{ij}]_{m\times q}$ as well as a weighted voting matrix $\mathbf{Z} = [Z_{ij}]_{m\times q}$ are specified by aggregating the labeling information of each neighboring example in $\mathcal{N}(\boldsymbol{x}_i')$:

$$\begin{aligned} &\forall\, 1 \le i \le m,\; 1 \le j \le q : \\ &\quad V_{ij} = \sum_{(\boldsymbol{x}_a', S_a)\in\mathcal{N}(\boldsymbol{x}_i')} [\![l_j \in S_a]\!] \\ &\forall\, 1 \le i \le m,\; 1 \le j \le q : \\ &\quad Z_{ij} = \sum_{(\boldsymbol{x}_a', S_a)\in\mathcal{N}(\boldsymbol{x}_i')} Y_{aj} \cdot [\![l_j \in S_a]\!] \cdot \omega_a \end{aligned} \tag{18}$$

Here, $[\![\pi]\!]$ returns 1 if predicate π holds and 0 otherwise. Conceptually, V_{ij} stores the number of neighboring examples of $\boldsymbol{x}_i'$ which take l_j as their candidate label. Obviously, $V_{ij} \le k$ holds for each element in $\mathbf{V}$. Furthermore, for the a-th neighboring example ($1 \le a \le k$), the voting weight is set as $\omega_a = k - a + 1$ [20, 52]. Therefore, based on current labeling confidence matrix $\mathbf{Y}$ and the voting weights, Z_{ij} consolidates the labeling confidence of l_j being the ground-truth label for each neighboring example.

Table 1. The pseudo-code of Delin.

```
Inputs:
    $\mathcal{D}$:   the PL training set $\{(\boldsymbol{x}_i, S_i) \mid 1 \le i \le m\}$ ($\mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{l_1, l_2, \ldots, l_q\}, \boldsymbol{x}_i \in \mathcal{X}, S_i \subseteq \mathcal{Y}$)
    $d'$:   the dimensionality of LDA-induced feature space
    mode:   the LDA mode (plain or kernelized) for dimensionality reduction
    $k$:   the number of nearest neighbors used for candidate label disambiguation
Outputs:
    $\mathcal{D}'$:   the transformed PL training set $\{(\boldsymbol{x}_i, S_i) \mid 1 \le i \le m\}$ in the LDA-induced feature space
Process:
Initialize the $m \times q$ labeling confidence matrix $\mathbf{Y}$ according to Eq.(1);
repeat
    if mode = plain then
        Specify the global mean vector $\boldsymbol{\mu}$ and the class-wise mean vector $\boldsymbol{\mu}_j$ $(1 \le j \le q)$ according to Eq.(3);
        Derive the total scatter matrix $\mathrm{S}_t$ and within-class scatter matrix $\mathrm{S}_w$ according to Eq.(4);
        Derive the between-class scatter matrix $\mathrm{S}_b$ according to Eq.(5);
        Form the LDA projection matrix $\mathbf{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_{d'}]$ with $\boldsymbol{w}_h$ $(1 \le h \le d')$ set to be the eigenvector w.r.t.
        the top-$h$ eigenvalue of $\mathrm{S}_w^{-1}\mathrm{S}_b$ satisfying $\boldsymbol{w}_h^\top \mathrm{S}_w \boldsymbol{w}_h = 1$;
        Set the transformed PL training set $\mathcal{D}' = \{(\boldsymbol{x}_i', S_i) \mid \boldsymbol{x}_i' = \mathbf{W}^\top \boldsymbol{x}_i, 1 \le i \le m\}$;
    else
        Set $\boldsymbol{\phi}^* = [\phi_1^*, \phi_2^*, \ldots, \phi_m^*]^\top$ and $\boldsymbol{\phi}_j = [\phi_{j1}, \phi_{j2}, \ldots, \phi_{jm}]^\top$ $(1 \le j \le q)$ according to Eq.(14) with the specified
        kernel function $\kappa(\cdot, \cdot)$;
        Derive $\Phi = \sum_{j=1}^q (\boldsymbol{\phi}_j - \boldsymbol{\phi}^*)(\boldsymbol{\phi}_j - \boldsymbol{\phi}^*)^\top$;
        Set $\mathbf{K} = [K_{ij}]_{m \times m}$ with $K_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and $\mathbf{H}_j$ $(1 \le j \le q)$ according to Eq.(16);
        Derive $\Omega = \sum_{j=1}^q \sum_{i=1}^m Y_{ij} \cdot \mathbf{K}\mathbf{H}_j\mathbf{K}^\top$;
        Form the coefficient matrix $\mathbf{A} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_{d'}]$ with $\boldsymbol{\alpha}_h$ $(1 \le h \le d')$ set to be the eigenvector w.r.t. the
        top-$h$ eigenvalue of $\Omega^{-1}\Phi$ satisfying $\boldsymbol{\alpha}_h^\top \Omega \boldsymbol{\alpha}_h = 1$;
        Set the transformed PL training set $\mathcal{D}' = \{(\boldsymbol{x}_i', S_i) \mid \boldsymbol{x}_i' = \mathbf{A}^\top \mathbf{K}(:, i), 1 \le i \le m\}$ ($\mathbf{K}(:, i)$ being the $i$-th column
        of $\mathbf{K}$);
    end if
    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 counting matrix $\mathbf{V}$ and weighted voting matrix $\mathbf{Z}$ according to Eq.(18);
    Specify the updated labeling confidence matrix $\mathbf{Y}'$ according to Eqs.(19)-(20);
    Set $\mathbf{Y} = \mathbf{Y}'$;
until convergence
```

Given the PL example $(\boldsymbol{x}_i', S_i)$ in LDA-induced feature space, we denote l_{i^*} as the candidate label in S_i which has largest weighted voting:[1]

$$l_{i^*} = \arg\max_{l_j \in S_i} \; Z_{ij} \tag{19}$$

[1] In case that there are more than one candidate label which have the same largest weighted voting, one of them will be randomly selected to instantiate l_{i^*}.

Then, we set the updated labeling confidence matrix $\mathbf{Y}'$ as follows:

$$\forall\ 1 \leq i \leq m,\ 1 \leq j \leq q: \tag{20}$$

$$Y'_{ij} = \begin{cases} [\![j = i^*]\!], & \text{if } |S_i| = 1 \\ \frac{V_{ii^*}}{k}, & \text{if } |S_i| > 1 \text{ and } j = i^* \\ \left(1 - \frac{V_{ii^*}}{k}\right)/(|S_i| - 1), & \text{if } |S_i| > 1 \text{ and } j \neq i^* \end{cases}$$

In other words, the labeling confidence for the candidate label with largest weighted voting (i.e. l_{i^*}) will be updated by referring to the counting statistic V_{ii^*}. Then, the remaining labeling confidences are shared among the candidate labels in S_i other than l_{i^*}.

Table 1 summarizes the complete procedure of Delin. Firstly, the labeling confidence matrix is initialized based on the candidate label assignment (Step 1). Then, an iterative procedure alternating between LDA dimensionality reduction (Steps 4-8 for *plain* mode, or Steps 10-15 for *kernelized* mode) and candidate label disambiguation (Steps 17-22) is conducted. Here, the iterative procedure terminates if the labeling confidence matrix does not change or the maximum number of iterations is reached.[2] Consequently, the transformed PL training set $\mathcal{D}'$ in the LDA-induced feature space can be utilized for follow-up model training.

4 EXPERIMENTS

4.1 Experimental Setup

To the best of our knowledge, Delin serves as the first approach towards dimensionality reduction for partial label data. To show the effectiveness of Delin (as well as its kernelized version Delin$^{\mathcal{K}}$), we investigate the performance of well-established partial label learning algorithms after coupling with the proposed dimensionality reduction approach. For any partial label learning algorithm $\mathcal{A}$, we use $\mathcal{A}$-Delin ($\mathcal{A}$-Delin$^{\mathcal{K}}$) to denote its coupling version with Delin (Delin$^{\mathcal{K}}$) which learns from partial label training examples in the LDA-induced feature space. Accordingly, to verify whether the proposed dimensionality reduction approach is effective in improving the generalization ability of partial label learning system, the performance of $\mathcal{A}$-Delin ($\mathcal{A}$-Delin$^{\mathcal{K}}$) trained on the transformed PL training set $\mathcal{D}'$ is compared against that of $\mathcal{A}$ trained on the original PL training set $\mathcal{D}$.

To perform thorough comparative studies, a total of four well-established partial label learning algorithms are employed to instantiate $\mathcal{A}$, each configured with parameters suggested in respective literatures:

- Pl-knn [20]: An instance-based partial label learning algorithm which learns from PL examples by making prediction on unseen instance via weighted kNN voting [suggested configuration: k=10].
- Pl-svm [31]: A maximum-margin partial label learning algorithm which learns from PL examples by maximizing the classification margin over candidate label set and non-candidate label set [suggested configuration: regularization parameter pool with $\{10^{-3}, \ldots, 10^{3}\}$].
- Pl-ecoc [53]: A transformation-based partial label learning algorithm which learns from PL examples by decomposing the PL learning problem into a number of binary learning problems via adapting the error-correcting output codes (ECOC) techniques [suggested configuration: ECOC coding length $\lceil 10 \cdot \log_2(q) \rceil$].

[2] In this paper, the maximum number of iterations is set to be 75 which suffices to yield stable performance for the proposed approach.

Table 2. Characteristics of the synthetic experimental 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* [36]
tmc2007	8,670	981	18	r = 1, 2, 3	*text anomaly detection* [38]
slashdot	3,142	1,079	19	r = 1, 2, 3	*text classification* [24]
amazon	1,500	1,326	50	r = 1, 2, 3	*authorship identification* [13]
DeliciousMIL	1,409	1,389	20	r = 1, 2, 3	*sentence labeling* [37]
bookmark	2,500	1,413	57	r = 1, 2, 3	*automatic tag suggestion* [23]
sports	9,120	1,738	19	r = 1, 2, 3	*human activity recognition* [1]
sector	6,412	6,104	105	r = 1, 2, 3	*text classification* [35]

- Ipal [52]: Another instance-based partial label learning algorithm which learns from PL examples by making prediction on unseen instance via adapting label propagation for graph-based disambiguation [suggested configuration: balancing parameter $\alpha = 0.95$].

For Delin, the two parameters d' (i.e. # features after LDA dimensionality reduction) and k (i.e. # nearest neighbors for candidate label disambiguation) as shown in Table 1 are set to be $\lceil thr \cdot \min(q, d) \rceil$ with $thr = 0.6$ and $k = 8$ respectively. Furthermore, linear kernel is utilized to fulfill the kernelized version Delin$^{\mathcal{K}}$.

In this paper, comparative studies are conducted on synthetic as well as real-world data sets. Ten-fold cross-validation is performed on each data set, and the detailed experimental results (mean classification accuracy with standard deviation) are reported in the following subsections.

4.2 Synthetic Data Sets

To generate synthetic PL data set, we follow the widely-used strategy [8, 9, 11, 17, 26, 50, 53] to derive PL examples from multi-class examples with controlling parameter r. Specifically, r controls the number of *false positive* labels in the candidate label set of PL example. Given a multi-class example $(\boldsymbol{x}_i, y_i)$, one PL example $(\boldsymbol{x}_i, S_i)$ can be generated by randomly adding r false positive labels $\Delta_r \subseteq \mathcal{Y} \setminus \{y_i\}$ into S_i along with the ground-truth label y_i, i.e. $S_i = \Delta_r \bigcup \{y_i\}$ ($|S_i| = r + 1$).

Characteristics of the synthetic data sets ($r \in \{1, 2, 3\}$) are summarized in Table 2, where each data set is roughly ordered based on its number of features.[3] Accordingly, detailed experimental results of each comparing algorithm are reported in Table 3. For partial label learning algorithm $\mathcal{A} \in$ {Pl-knn, Pl-svm, Pl-ecoc, Ipal}, both $\mathcal{A}$-Delin and $\mathcal{A}$-Delin$^{\mathcal{K}}$ are compared against $\mathcal{A}$ where the best classification accuracy is shown in boldface. Furthermore, to show whether the performance difference between $\mathcal{A}$-Delin ($\mathcal{A}$-Delin$^{\mathcal{K}}$) and $\mathcal{A}$ is significant, pairwise t-test at 0.05 significance level is conducted where the resulting win/tie/loss counts are reported in Table 4.

Based on the reported results on synthetic data sets, we can observe that:

- Among all the 96 cases (8 data sets × 3 settings of r × 4 PL learning algorithms; Table 3), coupling with the proposed dimensionality reduction approach (i.e. Delin and Delin$^{\mathcal{K}}$) would lead to better performance than the original partial label learning algorithm $\mathcal{A}$ in 98.9% cases. The only exception is on `mediamill` ($r = 1$; $\mathcal{A}$ = Ipal) which corresponds to the synthetic data set with least number of features under least number of false positive labels. Furthermore, $\mathcal{A}$-Delin$^{\mathcal{K}}$ achieves better performance than $\mathcal{A}$-Delin in 74 out of 96 cases.

[3] In Table 2, most multi-class data sets are derived from multi-label benchmark data sets [58] by retaining examples with only one relevant label.

Table 3. Classification accuracy (mean±std. deviation) of each comparing algorithm on controlled synthetic data sets ($r \in \{1, 2, 3\}$). For partial label learning algorithm $\mathcal{A} \in$ {Pl-knn, Pl-svm, Pl-ecoc, Ipal}, the performance of $\mathcal{A}$-Delin and $\mathcal{A}$-Delin$^{\kappa}$ are compared against that of $\mathcal{A}$ where the best performance on each data set is shown in boldface.

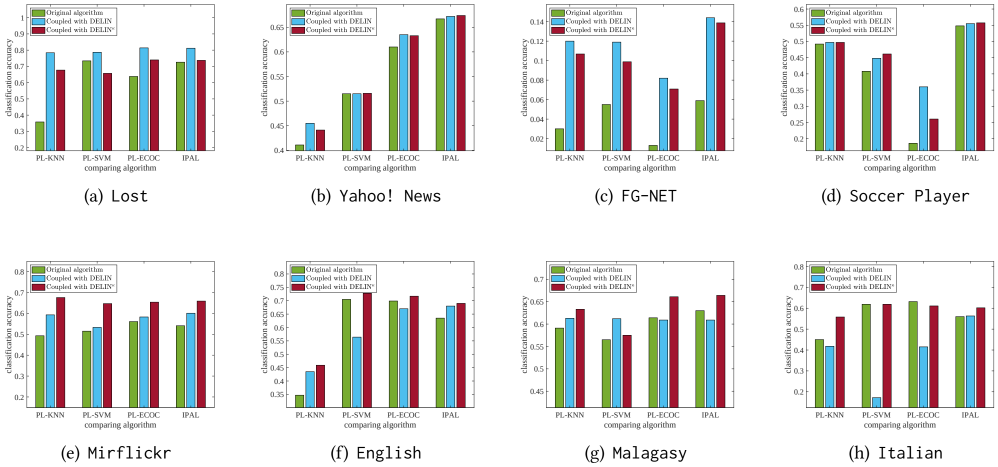
Comparing Algorithms	**Data Set**							
	mediamill	**tmc2007**	**slashdot**	**amazon**	**DeliciousMIL**	**bookmark**	**sports**	**se ctor**
				r = 1 (*one false positive label*)				
Pl-knn	0.637 ± 0.034	0.402 ± 0.012	0.163 ± 0.022	0.025 ± 0.033	0.033 ± 0.039	0.170 ± 0.026	0.288 ± 0.015	0.014 ± 0.005
Pl-knn-Delin	**0.688 ± 0.027**	0.654± 0.013	0.698 ± 0.033	**0.609 ± 0.048**	0.464 ± 0.043	0.536 ± 0.036	0.865± 0.014	0.530 ± 0.034
Pl-knn-Delin$^{\kappa}$	0.649 ± 0.019	**0.665± 0.014**	**0.705 ± 0.017**	0.594 ± 0.039	**0.523 ± 0.021**	**0.658 ± 0.025**	**0.929 ± 0.006**	**0.562 ± 0.030**
Pl-svm	0.495 ± 0.042	0.645± 0.021	0.595 ± 0.018	0.120 ± 0.026	0.036 ± 0.017	0.279 ± 0.029	0.677± 0.019	0.070 ± 0.012
Pl-svm-Delin	**0.600 ± 0.037**	0.666 ± 0.013	0.717±0.029	0.558 ± 0.038	0.354 ± 0.043	0.534 ± 0.037	0.709±0.013	**0.496 ±0.035**
Pl-svm-Delin$^{\kappa}$	0.527± 0.032	**0.675 ± 0.014**	**0.727±0.024**	**0.631 ± 0.054**	**0.471 ± 0.037**	**0.668 ± 0.030**	**0.726± 0.015**	**0.496 ± 0.035**
Pl-ecoc	0.592 ± 0.037	0.635±0.016	0.528± 0.033	0.065 ± 0.021	0.072 ± 0.038	0.352 ± 0.039	0.697±0.031	0.058 ± 0.012
Pl-ecoc-Delin	**0.666 ± 0.037**	0.669± 0.013	0.719±0.027	0.608 ± 0.046	0.464 ± 0.042	0.550 ± 0.033	0.851± 0.013	0.527 ± 0.033
Pl-ecoc-Delin$^{\kappa}$	0.598 ± 0.040	**681±0.013**	**0.729± 0.021**	**0.693 ±0.031**	**0.524 ± 0.021**	**0.683 ± 0.024**	**0.920 ± 0.007**	**0.559 ± 0.030**
Ipal	**0.642 ±0.020**	0.598±0.019	0.417±0.023	0.105 ± 0.062	0.062 ±0.017	0.309± 0.030	0.905± 0.009	0.144 ±0.015
Ipal-Delin	0.640 ±0.037	0.610±0.019	0.694±0.027	0.610 ± 0.048	0.463 ±0.044	0.550 ± 0.027	0.880±0.011	0.531 ±0.034
Ipal-Delin$^{\kappa}$	0.619 ±0.026	**0.624±0.015**	**0.709±0.031**	**0.663 ± 0.044**	**0.524 ±0.021**	**0.668± 0.032**	**0.943±0.005**	**0.563 ± 0.029**
				r = 2 (*two false positive labels*)				
Pl-knn	0.623 ±0.023	0.379±0.016	0.160±0.020	0.021±0.009	0.027 ±0.014	0.162 ±0.012	0.290±0.015	0.015 ± 0.007
Pl-knn-Delin	**0.665 ±0.036**	0.650±0.013	**0.668±0.018**	0.466 ±0.021	0.258±0.042	0.486 ±0.033	0.842± 0.018	0.392 ±0.022
Pl-knn-Delin$^{\kappa}$	0.628 ± 0.020	**0.659±0.007**	0.655±0.021	**0.589 ±0.040**	**0.365± 0.043**	**0.560±0.020**	**0.911 ±0.009**	**0.415 ± 0.025**
Pl-svm	0.490 ±0.041	0.631±0.039	0.575± 0.029	0.073 ± 0.021	0.035 ±0.019	0.261±0.019	0.640±0.015	0.054 ± 0.011
Pl-svm-Delin	**0.608 ± 0.016**	0.668±0.016	**0.687± 0.024**	0.438 ±0.023	0.220±0.038	0.504 ± 0.030	0.686±0.015	0.373± 0.022
Pl-svm-Delin$^{\kappa}$	0.536 ± 0.029	**0.677 ±0.009**	0.666±0.025	**0.639± 0.040**	**0.324 ± 0.039**	**0.585 ± 0.018**	**0.695± 0.018**	**0.381 ±0.023**
Pl-ecoc	0.514 ± 0.036	0.584±0.027	0.428±0.035	0.040 ±0.016	0.063 ± 0.034	0.284 ±0.035	0.601±0.037	0.036±0.009
Pl-ecoc-Delin	**0.598 ±0.039**	0.653±0.017	**0.688±0.023**	0.466 ± 0.022	0.253 ±0.039	0.495 ± 0.033	0.818±0.013	0.390±0.022
Pl-ecoc-Delin$^{\kappa}$	0.552 ±0.043	**0.661±0.013**	0.672±0.022	**0.649 ± 0.037**	**0.365 ± 0.044**	**0.572 ± 0.021**	**0.887±0.008**	**0.413 ± 0.026**
Ipal	0.592 ± 0.023	0.583±0.009	0.402± 0.025	0.088 ± 0.020	0.052 ± 0.011	0.304 ±0.018	0.901± 0.008	0.136 ±0.009
Ipal-Delin	**0.597 ± 0.030**	0.606± 0.018	**0.664±0.023**	0.468 ± 0.021	0.258 ±0.042	0.499 ± 0.038	0.863± 0.013	0.392± 0.022
Ipal-Delin$^{\kappa}$	0.565 ± 0.021	**0.613±0.012**	0.653± 0.021	**0.681 ± 0.061**	**0.368 ± 0.043**	**0.564 ± 0.017**	**0.922± 0.008**	**0.415 ± 0.025**
				r = 3 (*three false positive labels*)				
Pl-knn	0.598 ± 0.017	0.364±0.011	0.165± 0.030	0.021 ± 0.008	0.043 ± 0.022	0.140±0.012	0.292±0.021	0.017± 0.005
Pl-knn-Delin	**0.656 ±0.022**	0.627±0.013	**0.642±0.033**	0.347 ±0.027	0.198 ± 0.035	0.437 ±0.037	0.824± 0.012	0.295 ± 0.018
Pl-knn-Delin$^{\kappa}$	0.592 ± 0.019	**0.639±0.015**	0.610±0.033	**0.525 ±0.037**	**0.244 ±0.033**	**0.521 ± 0.037**	**0.896± 0.010**	**0.318 ± 0.019**
Pl-svm	0.471±0.039	0.619±0.035	0.562±0.038	0.055±0.019	0.038±0.020	0.247 ± 0.028	0.603±0.019	0.047± 0.008
Pl-svm-Delin	**0.602 ± 0.031**	0.659±0.018	**0.667±0.035**	0.309 ± 0.030	0.158 ±0.032	0.452 ± 0.040	0.641±0.021	0.273 ± 0.019
Pl-svm-Delin$^{\kappa}$	0.501 ±0.037	**0.669±0.019**	0.637±0.028	**0.587 ±0.038**	**0.214 ± 0.027**	**0.540 ± 0.041**	**0.675 ± 0.015**	**0.274±0.018**
Pl-ecoc	0.101 ± 0.024	0.568±0.021	0.373±0.039	0.031±0.017	0.063 ±0.036	0.203±0.043	0.492±0.043	0.020 ± 0.007
Pl-ecoc-Delin	**0.231 ±0.113**	0.576±0.033	**0.645±0.035**	0.346 ± 0.026	0.188 ±0.032	0.443 ±0.036	0.762±0.022	0.293 ± 0.017
Pl-ecoc-Delin$^{\kappa}$	0.109 ± 0.026	**0.594±0.040**	0.618± 0.032	**0.533 ± 0.037**	**0.244±0.033**	**0.539 ± 0.033**	**0.830± 0.018**	**0.315±0.021**
Ipal	0.525 ±0.024	0.557±0.016	0.373± 0.030	0.084 ± 0.024	0.044± 0.015	0.293 ± 0.042	0.892±0.009	0.133 ± 0.013
Ipal-Delin	**0.564 ± 0.026**	0.593±0.013	**0.639±0.038**	0.349 ± 0.027	0.197 ±0.036	0.447 ±0.032	0.840± 0.019	0.294 ±0.017
Ipal-Delin$^{\kappa}$	0.512 ± 0.028	**0.602±0.015**	0.606± 0.034	**0.644± 0.033**	**0.244± 0.033**	**0.539 ±0.035**	**0.911± 0.010**	**0.318 ±0.019**

Table 4. Win/tie/loss counts (pairwise t-test at 0.05 significance level) between $\mathcal{A}$-Delin ($\mathcal{A}$-Delin$^{\kappa}$) and $\mathcal{A}$ in terms of different number of false positive labels (r = 1, 2, 3).

	$\mathcal{A}$-Delin against $\mathcal{A}$				$\mathcal{A}$-Delin$^{\kappa}$ against $\mathcal{A}$			
	$\mathcal{A}$=Pl-knn	$\mathcal{A}$= Pl-svm	$\mathcal{A}$=Pl-ecoc	$\mathcal{A}$=Ipal	$\mathcal{A}$=Pl-knn	$\mathcal{A}$= Pl-svm	$\mathcal{A}$= Pl-ecoc	$\mathcal{A}$=Ipal
r = 1	8/0/0	8/0/0	8/0/0	6/1/1	8/0/0	8/0/0	8/0/0	7/0/1
r = 2	8/0/0	8/0/0	8/0/0	6/1/1	8/0/0	8/0/0	8/0/0	7/0/1
r = 3	8/0/0	8/0/0	8/0/0	7/0/1	7/0/1	8/0/0	8/0/0	7/0/1
In Total	**24/0/0**	**24/0/0**	**24/0/0**	**19/2/3**	**23/0/1**	**24/0/0**	**24/0/0**	**21/0/3**

- For Pl-knn, the coupling versions Pl-knn-Delin and Pl-knn-Delin$^{\kappa}$ significantly outperform Pl-knn in 100% and 95.8% cases respectively (Table 4). Specifically, on `tmc2007` where Pl-knn has the second highest classification accuracy, the classification accuracy has been improved with Delin by 0.252, 0.271 and 0.263 (with Delin$^{\kappa}$ by 0.263, 0.280 and 0.275) for r = 1, 2 and 3 respectively. For `sector` on which Pl-knn has the lowest predictive accuracy, the performance improvement is even more pronounced with Delin by an increase of 0.516, 0.377 and 0.278 (with Delin$^{\kappa}$ by an increase of 0.548, 0.400 and 0.301) for r = 1, 2 and 3 respectively. And in most cases, the effect of Delin^{K} is more significant than Delin. For example, on amazon with $r = 2$ and 3, the improvement of classification accuracy brought by Delin^{K} for Pl-knn is 0.123 and 0.178 higher than that of Delin respectively.

- For both Pl-svm and Pl-ecoc, their performance have been significantly improved by Delin and Delin^{K} in all cases (Table 4). On the five data sets with more than 1,300 features (i.e. amazon, DeliciousMIL, bookmark, sports and sector), out of the 30 statistical comparisons (2 PL learning algorithms x 5 data sets x 3 settings of r), the classification accuracy has been improved with Delin by more than 0.20 in 22 cases (with Delin^{K} by more than 0.20 in 25 cases). These results indicate that the benefits brought by Delin as well as Delin^{K} are rather noticeable under the challenging circumstances of high dimensionality.
- For Ipal, the coupling version Ipal-Delin is outperformed by Ipal on sports which has largest number of examples, and the other coupling version Ipal-Delin^{K} is outperformed by Ipal on mediamill which has least number of features and class labels (Table 3). Nonetheless, on the two data sets amazon and DeliciousMIL with least number of examples, the classification accuracy has been improved with Delin by more than 0.40, 0.20 and 0.15 (with Delin^{K} by more than 0.45, 0.30 and 0.20) for $r = 1$, 2 and 3 respectively. These results indicate that the benefits brought by Delin as well as Delin^{K} are rather noticeable under the challenging circumstances of insufficient training examples.



(a) Lost (b) Yahoo! News (c) FG-NET (d) Soccer Player

(e) Mirflickr (f) English (g) Malagasy (h) Italian

Fig. 1. Classification accuracy of each partial label learning algorithm on real-world data sets (*green bar*: original algorithm; *blue bar*: coupled with Delin; *red bar*: coupled with Delin^{K}).

4.3 Real-World Data Sets

Characteristics of the real-world partial label data sets are summarized in Table 5, which are collected from different task domains including FG-NET [32] for facial age estimation, Lost [11], Soccer Player [51] and Yahoo! News [18] for automatic face naming from images or videos, Mirflickr [19] for web image classification, and English[56], Malagasy[12], Italian[56] for part-of-speech (POS) tagging.[4] For *facial age estimation*, instances correspond to human faces with landmarks while candidate labels correspond to ages annotations given by crowdsourced labelers. For *automatic face naming*, instances correspond to faces cropped from an image or video frame while candidate labels correspond to names extracted from the associated captions or subtitles. For *web image classification*, instances correspond to web images while candidate labels correspond to annotations extracted from the web environment. For *POS tagging*, instances correspond to the target words with contextual features while candidate labels correspond to the part-of-speech tags that the target words may have.

On each real-world data set, the classification accuracy of each partial label learning algorithm before and after employing the proposed dimensionality reduction techniques is illustrated in Fig. 1. Furthermore, to show whether the performance difference between $\mathcal{A}$-Delin ($\mathcal{A}$-Delin$^{\mathcal{K}}$) and $\mathcal{A}$ is significant, pairwise t-test at 0.05 significance level is conducted where the win/tie/loss statistics are reported in Table 6.

Based on the reported results on real-world data sets, we can observe that:

- Out of the 32 statistical comparisons (4 PL learning algorithms x 8 data sets), the classification accuracy of partial label learning algorithm $\mathcal{A}$ has been significantly improved in 71.8% and 84.3% cases by employing Delin and Delin$^{\mathcal{K}}$ for dimensionality reduction respectively (Table 6). The six losses of $\mathcal{A}$-Delin against $\mathcal{A}$ take place on data sets English, Malagasy and Italian from the *POS tagging* task domain (Fig. 1(f)-(h)), while on the other task domains $\mathcal{A}$-Delin achieves superior or at least statistically comparable performance against $\mathcal{A}$. The only two losses of $\mathcal{A}$-Delin$^{\mathcal{K}}$ against $\mathcal{A}$ take place on data set Lost ($\mathcal{A}$ = Pl-svm) with least number of features and Italian ($\mathcal{A}$ = Pl-ecoc) with small average number of candidate labels, while on the other cases $\mathcal{A}$-Delin$^{\mathcal{K}}$ achieves superior or at least statistically comparable performance against $\mathcal{A}$.
- As shown in Fig. 1(a), the performance improvement of $\mathcal{A}$-Delin against $\mathcal{A}$ is apparently higher than that of $\mathcal{A}$-Delin$^{\mathcal{K}}$ on the Lost data set, which has least number of features and small number of class labels in the label space. This indicates that it would be preferred to utilize the proposed dimensionality reduction approach in linear mode (i.e. Delin) rather than kernelized mode (i.e. Delin$^{\mathcal{K}}$) for data sets with lower dimensionality and smaller label space.
- As shown in Fig. 1(c), the performance improvement of both $\mathcal{A}$-Delin and $\mathcal{A}$-Delin$^{\mathcal{K}}$ against $\mathcal{A}$ is rather pronounced on the FG-NET data set, which is challenging to handle with least number of examples but large average number of candidate labels. Impressively, the classification accuracy of each partial label learning algorithm has at least been doubled on FG-NET by coupling with Delin and Delin$^{\mathcal{K}}$. These results indicate that the benefits brought by Delin as well as Delin$^{\mathcal{K}}$ are rather noticeable under the challenging circumstances of insufficient training examples and high rate of false positive labels.
- As shown in Fig. 1(e), the performance improvement of $\mathcal{A}$-Delin$^{\mathcal{K}}$ against $\mathcal{A}$ is apparently higher than that of $\mathcal{A}$-Delin on the Mirflickr data set, which has largest number of features. This indicates that it would be preferred to utilize the proposed dimensionality reduction approach in kernelized mode (i.e. Delin$^{\mathcal{K}}$) rather than linear mode (i.e. Delin) for data sets with high dimensionality.

4.4 Further Analysis

4.4.1 Effect of Reduced Dimensionality. For the proposed dimensionality reduction approach (Table 1), the key parameter corresponds to the number of retained features in the LDA-induced feature space (i.e. d'). Following the

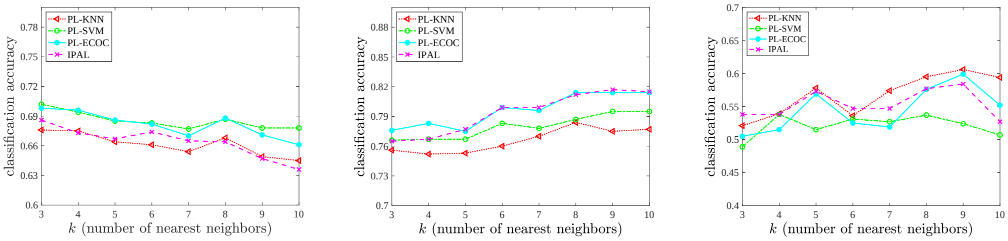
[4] Data sets available at: http://palm.seu.edu.cn/zhangml/Resources. htm#partial_data

Table 5. Characteristics of the real-world experimental data sets.

Data Set	# Examples	# Features	# Class Labels	average # Candidate Labels	Task Domain
Lost	1,122	108	16	2.23	*automatic face naming* [11]
Yahoo! News	22,991	163	219	1.91	*automatic face naming* [18]
FG-NET	1,002	262	78	7.48	*facial age estimation* [32]
Soccer Player	17,472	279	171	2.09	*automatic face naming* [51]
Mirflickr	2,780	1,536	14	2.76	*web image classification* [19]
English	24,000	300	45	1.19	*POS tagging* [56]
Malagasy	5,303	384	44	8.35	*POS tagging* [56]
Italian	21,878	519	90	1.60	*POS tagging*[56]

Table 6. Win/tie/loss statistics (pairwise t-test at 0.05 significance level) between $\mathcal{A}$-Delin ($\mathcal{A}$-Delin^{K}) and $\mathcal{A}$ on real-world data sets.

Data Set	$\mathcal{A}$-Delin against $\mathcal{A}$				$\mathcal{A}$-Delin^{K} against $\mathcal{A}$			
	$\mathcal{A}$=Pl-knn	$\mathcal{A}$= Pl-svm	$\mathcal{A}$=Pl-ecoc	$\mathcal{A}$=Ipal	$\mathcal{A}$=Pl-knn	$\mathcal{A}$= Pl-svm	$\mathcal{A}$= Pl-ecoc	$\mathcal{A}$=Ipal
Lost	win	win	win	win	win	loss	win	win
Yahoo! News	win	tie	win	win	tie	win	win	tie
FG-NET	win	win	win	win	win	win	win	win
Soccer Player	tie	win	win	win	win	win	win	win
Mirflickr	win	win	win	win	win	win	win	win
English	win	loss	tie	win	win	win	win	win
Malagasy	win	win	loss	loss	win	win	win	win
Italian	loss	loss	loss	win	win	tie	loss	win
In Total	**6/1/1**	**5/1/2**	**5/1/2**	**7/0/1**	**7/1/0**	**6/1/1**	**7/0/1**	**7/1/0**



(a) `slashdot` ($r = 2$)

(b) `Lost`

(c) `Mirflickr`

Fig. 2. Classification accuracy of $\mathcal{A}$-Delin ($\mathcal{A}$ ∈{Pl-knn, Pl-svm, Pl-ecoc, Ipal}) changes as the number of nearest neighbors used for candidate label disambiguation (i.e. k) increases from 3 to 10 with step-size 1. (a) synthetic data set `slashdot` ($r = 2$); (b) real-world data set `Lost`; (c) real-world data set `Mirflickr`.

common practice of multi-class classification with LDA [16, 30], we set $d' = \lceil thr \cdot \min(q, d) \rceil$ with $thr \in (0, 1)$ which is less than q (# class labels) as well as d (# original features).

Tables 7 and 8 report the detailed experimental results of coupling Delin and Delin^{K} with each partial label learning algorithm on all real-world data sets respectively with varying number of retained features. Here, thr increases from 0.5 to 0.9 with step-size 0.1 and the best performance across different values of thr is shown in boldface. As shown in Tables 7 and 8, the performance of each partial label learning algorithm coupled with Delin or Delin^{K} fluctuates moderately as the value of thr changes. Specifically, there is no single value of thr which can consistently lead to best

Table 7. Classification accuracy of $\mathcal{A}$-Delin ($\mathcal{A}$ ∈{Pl-knn, Pl-svm, Pl-ecoc, Ipal}) changes as the number of retained features varies ($d' = \lceil thr \cdot \min(q, d) \rceil$ with thr increasing from 0.5 to 0.9 with step-size 0.1). On each data set, the best performance across different values of thr is shown in boldface. For reference purpose, the classification accuracy of $\mathcal{A}$ on the original feature space is also shown in the lower part of the table.

Data Set	*thr*	# Retained Features	Pl-knn-Delin	Pl-svm-Delin	Pl-ecoc-Delin	Ipal-Delin
Lost	0.5	8	0.790±0.050	0.790±0.051	0.794±0.046	0.792±0.049
	0.6	10	0.784±0.031	0.787±0.035	0.814±0.046	0.812±0.046
	0.7	12	0.808±0.046	0.813±0.046	0.842±0.050	0.833±0.051
	0.8	13	**0.823±0.045**	**0.822±0.044**	**0.845±0.043**	**0.858±0.051**
	0.9	15	0.790±0.027	0.790±0.032	0.819±0.039	0.823±0.039
Yahoo! News	0.5	82	**0.475±0.006**	0.509±0.008	**0.639±0.007**	0.671±0.005
	0.6	98	0.455±0.009	0.515±0.010	0.635±0.007	**0.672±0.006**
	0.7	115	0.437±0.007	0.517±0.011	0.628±0.007	0.671±0.004
	0.8	131	0.424±0.004	**0.518±0.009**	0.621±0.008	0.667±0.007
	0.9	147	0.413±0.005	**0.518±0.009**	0.615±0.009	0.666±0.005
FG-NET	0.5	39	**0.128±0.032**	0.115±0.030	0.076±0.028	0.143±0.036
	0.6	47	0.120±0.011	0.119±0.027	**0.082±0.035**	**0.144±0.037**
	0.7	55	0.090±0.031	0.116±0.036	0.067±0.029	0.114±0.040
	0.8	63	0.090±0.023	**0.122±0.031**	0.079±0.032	0.128±0.016
	0.9	71	0.074±0.025	0.119±0.031	0.066±0.029	0.132±0.019
Soccer Player	0.5	86	**0.497±0.013**	0.445±0.027	0.323±0.062	**0.556±0.015**
	0.6	103	**0.497±0.012**	0.448±0.033	**0.360±0.054**	0.555±0.012
	0.7	120	0.494±0.014	0.449±0.043	0.288±0.072	0.554±0.013
	0.8	137	0.493±0.013	**0.450±0.039**	0.297±0.065	0.554±0.013
	0.9	154	0.494±0.014	0.435±0.049	0.287±0.074	0.552±0.013
Mirflickr	0.5	7	0.579±0.077	0.504±0.159	0.507±0.132	0.538±0.099
	0.6	9	**0.593±0.011**	0.533±0.134	**0.583±0.118**	**0.601±0.115**
	0.7	10	0.523±0.117	0.543±0.100	0.526±0.113	0.534±0.105
	0.8	12	0.501±0.120	0.554±0.097	0.512±0.126	0.513±0.122
	0.9	13	0.499±0.106	**0.555±0.085**	0.523±0.101	0.513±0.106
English	0.5	23	**0.438±0.039**	0.545±0.021	0.660±0.034	**0.684±0.029**
	0.6	27	0.434±0.041	0.536±0.029	0.663±0.025	0.682±0.025
	0.7	32	0.432±0.040	0.532±0.026	0.677±0.021	0.681±0.026
	0.8	36	0.433±0.039	**0.571±0.041**	**0.678±0.027**	0.680±0.026
	0.9	41	0.423±0.040	0.525±0.019	0.675±0.028	0.672±0.025
Malagasy	0.5	22	0.696±0.032	0.628±0.085	**0.713±0.025**	**0.726±0.035**
	0.6	27	**0.713±0.028**	**0.667±0.063**	0.706±0.031	0.723±0.023
	0.7	31	0.690±0.031	0.633±0.051	0.701±0.024	0.696±0.024
	0.8	36	0.632±0.027	0.604±0.052	0.597±0.024	0.622±0.024
	0.9	40	0.611±0.043	0.645±0.070	0.604±0.029	0.619±0.025
Italian	0.5	45	**0.459±0.039**	**0.343±0.021**	**0.522±0.030**	**0.592±0.021**
	0.6	54	0.428±0.039	0.218±0.021	0.495±0.034	0.569±0.021
	0.7	63	0.437±0.039	0.200±0.021	0.479±0.028	0.578±0.021
	0.8	72	0.417±0.039	0.195±0.021	0.463±0.021	0.573±0.021
	0.9	81	0.384±0.040	0.133±0.021	0.422±0.033	0.553±0.021
		# Original Features	Pl-knn	Pl-svm	Pl-ecoc	Ipal
Lost	-	108	0.358±0.029	0.734±0.004	0.638±0.051	0.726±0.041
Yahoo! News	-	163	0.411±0.005	0.515±0.001	0.610±0.009	0.667±0.005
FG-NET	-	262	0.030±0.019	0.055±0.024	0.013±0.015	0.059±0.019
Soccer Player	-	279	0.492±0.014	0.408±0.043	0.186±0.064	0.548±0.014
Mirflickr	-	1,536	0.496±0.127	0.515±0.127	0.561±0.013	0.541±0.129
English	-	300	0.347±0.036	0.705±0.025	0.699±0.027	0.635±0.027
Malagasy	-	384	0.591±0.039	0.565±0.060	0.614±0.031	0.630±0.038
Italian	-	519	0.450±0.019	0.619±0.023	0.632±0.032	0.560±0.031

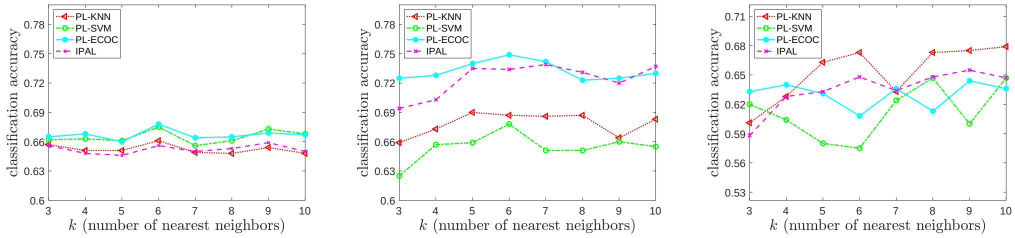
performance. Therefore, we have fixed the value of *thr* to be 0.6 for comparative studies while Delin and Delin$^{\kappa}$ may lead to further performance improvement by fine-tuning the value of *thr* on training set.

4.4.2 Effect of kNN-based Disambiguation. As shown in Table 1, another parameter for the proposed dimensionality reduction approach corresponds to the number of nearest neighbors used for candidate label disambiguation (i.e. k).

Table 8. Classification accuracy of $\mathcal{A}$-Delin$^{\mathcal{K}}$ ($\mathcal{A}$ $\in$\{Pl-knn, Pl-svm, Pl-ecoc, Ipal\}) changes as the number of retained features varies ($d' = \lceil thr \cdot \min(q, d) \rceil$ with thr increasing from 0.5 to 0.9 with step-size 0.1). On each data set, the best performance across different values of thr is shown in boldface. For reference purpose, the classification accuracy of $\mathcal{A}$ on the original feature space is also shown in the lower part of the table.

Data Set	*thr*	# Retained Features	Pl-knn-Delin$^{\mathcal{K}}$	Pl-svm-Delin$^{\mathcal{K}}$	Pl-ecoc-Delin$^{\mathcal{K}}$	Ipal-Delin$^{\mathcal{K}}$
Lost	0.5	8	0.674±0.049	0.639±0.051	0.714±0.073	0.689±0.057
	0.6	10	0.677±0.051	0.657±0.048	0.739±0.059	0.737±0.036
	0.7	12	**0.679±0.049**	0.668±0.062	0.739±0.065	0.749±0.054
	0.8	13	0.672±0.057	**0.682±0.048**	**0.748±0.052**	0.753±0.052
	0.9	15	0.663±0.057	0.672±0.062	0.745±0.054	**0.759±0.043**
Yahoo! News	0.5	82	**0.465±0.008**	0.508±0.013	**0.640±0.004**	0.666±0.005
	0.6	98	0.441±0.008	0.515±0.009	0.632±0.006	**0.673±0.006**
	0.7	115	0.426±0.009	0.518±0.010	0.628±0.006	0.669±0.006
	0.8	131	0.412±0.005	**0.520±0.008**	0.621±0.008	0.666±0.005
	0.9	147	0.403±0.006	0.519±0.009	0.613±0.007	0.665±0.006
FG-NET	0.5	39	**0.122±0.020**	**0.130±0.031**	0.064±0.032	**0.139±0.021**
	0.6	47	0.106±0.017	0.098±0.035	**0.070±0.030**	0.138±0.049
	0.7	55	0.102±0.025	0.114±0.022	0.055±0.027	0.111±0.029
	0.8	63	0.083±0.039	0.106±0.024	0.067±0.035	0.122±0.029
	0.9	71	0.099±0.034	0.112±0.022	0.059±0.028	0.124±0.031
Soccer Player	0.5	86	**0.499±0.013**	0.444±0.033	**0.276±0.034**	**0.557±0.011**
	0.6	103	0.496±0.013	**0.461±0.010**	0.260±0.065	**0.557±0.016**
	0.7	120	0.495±0.013	0.449±0.037	0.247±0.062	0.554±0.011
	0.8	137	0.495±0.013	0.448±0.041	0.224±0.075	0.553±0.013
	0.9	154	0.493±0.013	0.439±0.036	0.248±0.071	0.553±0.011
Mirflickr	0.5	7	**0.591±0.065**	0.491±0.152	**0.574±0.089**	0.541±0.090
	0.6	9	0.586±0.077	0.496±0.145	0.508±0.144	**0.573±0.081**
	0.7	10	0.506±0.108	**0.556±0.083**	0.482±0.130	0.516±0.107
	0.8	12	0.472±0.093	0.546±0.067	0.506±0.091	0.490±0.094
	0.9	13	0.478±0.092	0.547±0.073	0.486±0.080	0.492±0.093
English	0.5	23	0.455±0.042	0.703±0.023	0.698±0.030	**0.695±0.026**
	0.6	27	**0.459±0.041**	**0.728±0.025**	0.719±0.027	**0.695±0.026**
	0.7	32	0.456±0.041	0.727±0.027	**0.728±0.023**	0.694±0.028
	0.8	36	0.451±0.041	0.722±0.029	**0.728±0.027**	0.694±0.026
	0.9	41	0.450±0.043	0.722±0.029	0.725±0.024	0.688±0.026
Malagasy	0.5	22	**0.663±0.039**	0.534±0.053	**0.683±0.022**	**0.683±0.026**
	0.6	27	0.661±0.035	0.596±0.086	0.668±0.026	0.669±0.036
	0.7	31	0.648±0.035	0.570±0.055	0.559±0.023	0.668±0.030
	0.8	36	0.639±0.035	**0.617±0.061**	0.585±0.030	0.656±0.033
	0.9	40	0.647±0.031	0.594±0.072	0.636±0.027	0.679±0.032
Italian	0.5	45	**0.563±0.022**	0.660±0.030	0.643±0.015	0.597±0.030
	0.6	54	0.557±0.022	0.660±0.042	0.644±0.027	**0.639±0.029**
	0.7	63	0.552±0.021	**0.678±0.058**	**0.645±0.030**	0.621±0.028
	0.8	72	0.545±0.023	0.676±0.053	0.644±0.048	0.601±0.030
	0.9	81	0.543±0.025	0.650±0.046	0.644±0.032	0.615±0.038
		# Original Features	Pl-knn	Pl-svm	Pl-ecoc	Ipal
Lost	-	108	0.358±0.029	0.734±0.004	0.638±0.051	0.726±0.041
Yahoo! News	-	163	0.411±0.005	0.515±0.001	0.610±0.009	0.667±0.005
FG-NET	-	262	0.030±0.019	0.055±0.024	0.013±0.015	0.059±0.019
Soccer Player	-	279	0.492±0.014	0.408±0.043	0.186±0.064	0.548±0.014
Mirflickr	-	1,536	0.496±0.127	0.515±0.127	0.561±0.013	0.541±0.129
English	-	300	0.347±0.036	0.705±0.025	0.699±0.027	0.635±0.027
Malagasy	-	384	0.591±0.039	0.565±0.060	0.614±0.031	0.630±0.038
Italian	-	519	0.450±0.019	0.619±0.023	0.632±0.032	0.560±0.031

For illustrative purpose, Figs. 2 and 3 show how the performance of each partial label learning algorithm coupled with Delin and Delin$^{\mathcal{K}}$ changes respectively as k increases from 3 to 10 with step-size 1 on three data sets. As shown in Figs. 2 and 3, the performance of each partial label learning algorithm coupled with Delin or Delin$^{\mathcal{K}}$ is relatively stable as the value of k varies. Therefore, we have fixed the value of k to be 8 for comparative studies.



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

Fig. 3. Classification accuracy of $\mathcal{A}$-Delin$^{\mathcal{K}}$ ($\mathcal{A}$ ∈{Pl-knn, Pl-svm, Pl-ecoc, Ipal}) changes as the number of nearest neighbors used for candidate label disambiguation (i.e. k) increases from 3 to 10 with step-size 1. (a) synthetic data set `slashdot` ($r = 2$); (b) real-world data set `Lost`; (c) real-world data set `Mirflickr`.

5 CONCLUSION

In this paper, an extension to our earlier work [45] is presented which investigates the problem of dimensionality reduction for partial label learning. The proposed Delin approach enables LDA with the ability of dealing with partial label data in an iterative manner, which alternates between optimizing the projection matrix of LDA with disambiguation-guided labeling confidences and kNN-based candidate label disambiguation in the projected feature space. Comprehensive experimental studies over synthetic as well as real-world data sets clearly show that the generalization performance of well-established partial label learning algorithms can be significantly improved by coupling with Delin in either linear or kernelized mode.

Delin serves as an initial attempt towards partial label dimensionality reduction, it is important to explore other ways of enhancing partial label learning algorithms with feature manipulation techniques [4, 40]. Other than the iterative procedure, it is also interesting to investigate non-alternating procedure which can jointly perform dimensionality reduction and candidate label disambiguation.

REFERENCES

[1] K. Altun and B. Barshan. 2010. Human activity recognition using inertial/magnetic sensor units. In *Proceedings of the 1st International Conference on Human Behavior Understanding*. Istanbul, Turkey, 38–51.

[2] J. Amores. 2013. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence* 201 (2013), 81–105.

[3] F. Briggs, X. Z. Fern, and R. Raich. 2012. 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, 534–542.

[4] P. Bugata and P. Drotar. 2020. On some aspects of minimum redundancy maximum relevance feature selection. *Science China Information Sciences* 63, 1 (2020), Article 112103.

[5] M.-A. Carbonneaua, V. Cheplyginabc, E. Granger, and G. Gagnon. 2018. Multiple instance learning: A survey of problem characteristics and applications. *Pattern Recognition* 77 (2018), 329–353.

[6] J. Chai, I. W. Tsang, and W. Chen. 2020. Large margin partial label machine. *IEEE Transactions on Neural Networks and Learning Systems* 31, 7 (2020), 2594–2608.

[7] O. Chapelle, B. Schölkopf, and A. Zien (Eds.). 2006. *Semi-Supervised Learning*. MIT Press, Cambridge, MA.

[8] C.-H. Chen, V. M. Patel, and R. Chellappa. 2018. Learning from ambiguously labeled face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 7 (2018), 1653–1667.

[9] Y.-C. Chen, V. M. Patel, R. Chellappa, and P. J. Phillips. 2014. Ambiguously labeled learning using dictionaries. *IEEE Transactios on Information Forensics and Security* 9, 12 (2014), 2076–2088.

[10] T. Cour, B. Sapp, C. Jordan, and B. Taskar. 2009. Learning from ambiguously labeled images. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. Miami, FL, 919–926.

[11] T. Cour, B. Sapp, and B. Taskar. 2011. Learning from partial labels. *Journal of Machine Learning Research* 12, May (2011), 1501–1536.

[12] Garrette Dan and Jason Baldridge. 2013. Learning a Part-of-Speech Tagger from Two Hours of Annotation. *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies* 38, 2 (2013), 129–134.

[13] D. Dheeru and E. Karra Taniskidou. 2017. UCI Machine Learning Repository. (2017). http://archive.ics.uci.edu/ml

[14] L. Feng and B. An. 2018. Leveraging latent label distributions for partial label learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. Stockholm, Sweden, 2107–2113.

[15] L. Feng and B. An. 2019. Partial label learning by semantic difference maximization. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*. Macau, China, 2294–2300.

[16] K. Fukunaga. 2013. *Introduction to Statistical Pattern Recognition* (2nd edition ed.). Academic Press, Cambridge, MA.

[17] C. Gong, T. Liu, Y. Tang, J. Yang, J. Yang, and D. Tao. 2018. A regularization approach for instance-based superset label learning. *IEEE Transactions on Cybernetics* 48, 3 (2018), 967–978.

[18] M. Guillaumin, J. Verbeek, and C. Schmid. 2010. 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.). Springer, Berlin, 634–647.

[19] M. J. Huiskes and M. S. Lew. 2008. The MIR Flickr Retrieval Evaluation. In *Proceedings of the 1st ACM International Conference on Multimedia Information Retrieval*. Vancouver, Canada, 39–43.

[20] E. Hüllermeier and J. Beringer. 2006. Learning from ambiguously labeled examples. *Intelligent Data Analysis* 10, 5 (2006), 419–439.

[21] L. Jie and F. Orabona. 2010. Learning from candidate labeling sets. In *Advances in Neural Information Processing Systems 23*, J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta (Eds.). MIT Press, Cambridge, MA, 1504–1512.

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

[23] I. Katakis, G. Tsoumakas, and I. Vlahavas. 2008. Multilabel Text Classification for Automated Tag Suggestion. In *Proceedings of the ECML/PKDD 2008 Discovery Challenge*. Antwerp, Belgium.

[24] J. Leskovec, K. Lang, A. Dasgupta, and M. Mahoney. 2009. Community Structure in Large Networks: Natural Cluster Sizes and the Absence of Large Well-Defined Clusters. *Internet Mathematics* 6, 1 (2009), 29–123.

[25] X.-C. Li, D.-C. Zhan, J.-Q. Yang, and Y. Shi. 2021. Deep multiple instance selection. *Science China Information Sciences* 64, 3 (2021), Article 130102.

[26] L. Liu and T. Dietterich. 2012. 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.). MIT Press, Cambridge, MA, 557–565.

[27] J. Lv, M. Xu, L. Feng, G. Niu, X. Geng, and M. Sugiyama. 2020. Progressive identification of true labels for partial-label learning. In *Proceedings of the 37th International Conference on Machine Learning*. Virtual Conference, 6500–6510.

[28] G. Lyu, S. Feng, T. Wang, and C. Lang. 2021, in press. A Self-Paced Regularization Framework for Partial-Label Learning. *IEEE Transactions on Cybernetics* (2021, in press).

[29] G. Lyu, S. Feng, T. Wang, C. Lang, and Y. Li. 2021. GM-PLL: Graph matching based partial label learning. *IEEE Transactions on Knowledge and Data Engineering* 33, 2 (2021), 521–535.

[30] G. J. McLachlan. 2004. *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley & Sons, Inc., Hoboken, NJ.

[31] N. Nguyen and R. Caruana. 2008. Classification with partial labels. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Las Vegas, NV, 381–389.

[32] G. Panis and A. Lanitis. 2015. An overview of research activities in facial age estimation using the FG-NET aging database. In *Lecture Notes in Computer Science 8926*, C. Rother L. Agapito, M. M. Bronstein (Ed.). Springer, Berlin, 737–750.

[33] X. Ren, W. He, M. Qu, L. Huang, H. Ji, and J. Han. 2016. AFET: Automatic fine-grained entity typing by hierarchical partial-label embedding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*. Austin, TX, 1369–1378.

[34] X. Ren, W. He, M. Qu, C. R. Voss, H. Ji, and J. Han. 2016. Label noise reduction in entity typing by heterogeneous partial-label embedding. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco, CA, 1825–1834.

[35] J. D. M. Rennie and R. Rifkin. 2001. *Improving multiclass text classification with the support vector machines*. Technical Report AIM-2001-026. Artificial Intelligence Laboratory, Massachusetts Institute of Technology.

[36] C. G. M. Snoek, M. Worring, J. C. van Gemert, J.-M. Geusebroek, and A. W. M. Smeulders. 2006. 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, 421–430.

[37] H. Soleimani and D. J. Miller. 2016. 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, 105–114.

[38] A. N. Srivastava and B. Zane-Ulman. 2005. Discovering recurring anomalies in text reports regarding complex space systems. In *Proceedings of the 2005 IEEE Aerospace Conference*. Big Sky, MT.

[39] K. Sun, Z. Min, and J. Wang. 2019. PP-PLL: Probability Propagation for Partial Label Learning. In *Lecture Notes in Computer Science 11907*, U. Brefeld, E. Fromont, A. Hotho, A. Knobbe, M. Maathuis, and C. Robardet (Eds.). Springer, Berlin, 123–137.

[40] Y.-P. Sun and M.-L. Zhang. 2021. Compositional metric learning for multi-label classification. *Frontiers of Computer Science* 15, 5 (2021), Article 155320.

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

[42] D.-B. Wang, L. Li, and M.-L. Zhang. 2019. Adaptive graph guided disambiguation for partial label learning. In *Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. Anchorage, AK, 83–91.

[43] W. Wang and M.-L. Zhang. 2020. Semi-supervised partial label learning via confidence-rated margin maximization. In *Advances in Neural Information Processing Systems 33*, H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin (Eds.). MIT Press, Cambridge, MA, 6982–6993.

[44] Y. Wang, J. Han, Y. Shen, and H. Xue. 2021. Pointwise manifold regularization for semi-supervised learning. *Frontiers of Computer Science* 15, 1 (2021), Article 151303.

[45] J.-H. Wu and M.-L. Zhang. 2019. Disambiguation enabled linear discriminant analysis for partial label dimensionality reduction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. Anchorage, AK, 416–424.

[46] X. Wu and M.-L. Zhang. 2018. Towards enabling binary decomposition for partial label learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. Stockholm, Sweden, 2868–2974.

[47] M. Xu and L.-Z. Guo. 2021. Learning from group supervision: The impact of supervision deficiency on multi-label learning. *Science China Information Sciences* 64, 3 (2021), Article 130101.

[48] N. Xu, J. Lv, and X. Geng. 2019. Partial label learning via label enhancement. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*. Honolulu, HI, 5557–5564.

[49] Y. Yan and Y. Guo. 2020. Partial label learning with batch label correction. In *Proceedings of the 34th AAAI Conference on Artificial Intelligence*. New York, NY, 6575–6582.

[50] F. Yu and M.-L. Zhang. 2017. Maximum margin partial label learning. *Machine Learning* 106, 4 (2017), 573–593.

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

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

[53] M.-L. Zhang, F. Yu, and C.-Z. Tang. 2017. Disambiguation-free partial label learning. *IEEE Transactions on Knowledge and Data Engineering* 29, 10 (2017), 2155–2167.

[54] M.-L. Zhang and Z.-H. Zhou. 2014. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering* 26, 8 (2014), 1819–1837.

[55] S. Zhao, P. Ni, H. Chen, C. Li, and Z. Dai. 2021. Partial label learning via conditional-label-aware disambiguation. *Journal of Computer Science and Technology* 36, 3 (2021), 590–605.

[56] D. Zhou, Z. Zhang, M.-L. Zhang, and Y. He. 2018. Weakly supervised POS tagging without disambiguation. *ACM Transactions on Asian and Low-Resource Language Information Processing* 17, 4 (2018), Article 35.

[57] Z.-H. Zhou. 2018. A brief introduction to weakly supervised learning. *National Science Review* 5, 1 (2018), 44–53.

[58] Z.-H. Zhou and M.-L. Zhang. 2017. Multi-label learning. In *Encyclopedia of Machine Learning and Data Mining*, C. Sammut and G. I. Webb (Eds.). Springer, Berlin, 875–881.

[59] X. Zhu and A. B. Goldberg. 2009. Introduction to semi-supervised learning. In *Synthesis Lectures to Artificial Intelligence and Machine Learning*, R. J. Brachman and T. G. Dietterich (Eds.). Morgan & Claypool Publishers, San Francisco, CA, 1–130.