

Adaptive Graph Guided Disambiguation for Partial Label Learning

Deng-Bao Wang
College of Computer and
Information Science, Southwest
University, Chongqing 400715, China
dengbaowang@gmail.com

Li Li
College of Computer and
Information Science, Southwest
University, Chongqing 400715, China
lily@swu.edu.cn

Min-Ling Zhang*
¹School of Computer Science and
Engineering, Southeast University,
Nanjing 210096, China
²Key Laboratory of Computer
Network and Information Integration
(Southeast University), Ministry of
Education, China
zhangml@seu.edu.cn

ABSTRACT

Partial label learning aims to induce a multi-class classifier from training examples where each of them is associated with a set of candidate labels, among which only one is the ground-truth label. The common strategy to train predictive model is *disambiguation*, i.e. differentiating the modeling outputs of individual candidate labels so as to recover ground-truth labeling information. Recently, feature-aware disambiguation was proposed to generate different labeling confidences over candidate label set by utilizing the graph structure of feature space. However, the existence of noise and outliers in training data makes the similarity derived from original features less reliable. To this end, we proposed a novel approach for partial label learning based on *adaptive graph guided disambiguation* (PL-AGGD). Compared with fixed graph, adaptive graph could be more robust and accurate to reveal the intrinsic manifold structure within the data. Moreover, instead of the two-stage strategy in previous algorithms, our approach performs label disambiguation and predictive model training simultaneously. Specifically, we present a unified framework which jointly optimizes the ground-truth labeling confidences, similarity graph and model parameters to achieve strong generalization performance. Extensive experiments show that PL-AGGD performs favorably against state-of-the-art partial label learning approaches.

CCS CONCEPTS

• **Computing methodologies** → **Machine learning**; • **Information systems** → *Data mining*.

KEYWORDS

Partial label learning; Adaptive graph; Disambiguation

*Corresponding author

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

KDD '19, August 4–8, 2019, Anchorage, AK, USA

© 2019 Association for Computing Machinery.

ACM ISBN 978-1-4503-6201-6/19/08...\$15.00

<https://doi.org/10.1145/3292500.3330840>

ACM Reference Format:

Deng-Bao Wang, Li Li, and Min-Ling Zhang. 2019. Adaptive Graph Guided Disambiguation for Partial Label Learning. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3292500.3330840>

1 INTRODUCTION

Multi-class classification is an important problem in various real-world applications, such as natural language processing, computer vision, natural language processing, web advertising, etc [8, 30]. In ordinary multi-class classification problems, each training example is associated with a label which specifies the class which the example belongs to. Although learning from this kind of supervised datasets is effective, collecting labeled data is costly and thus corresponds to a critical bottleneck in real-world classification tasks. For this reason, learning from less expensive labeled data has been extensively studied in the last decades [7, 14, 20, 32].

Partial label learning [4, 11, 18, 22, 28] aims to induce a multi-class classifier from training examples where each of them is associated with a set of candidate labels, among which only one is the ground-truth label. This learning problem is motivated in domains in which a large number of ambiguously labeled examples are available while it is expensive to acquire explicit labels, and arises in many real-world tasks such as automatic image annotation [6, 25], web mining [17], ecoinformatics [16], etc.

Formally speaking, let $\mathcal{X} = \mathbb{R}^d$ be the d dimensional feature space and $\mathcal{Y} = \{0, 1\}^q$ be the label space with q possible class labels. Suppose we have the partial label training set $\mathcal{D} = \{(x_i, S_i) \mid 1 \leq i \leq m\}$ where $x_i \in \mathcal{X}$ is a d dimensional feature vector and S_i is the corresponding candidate label set. The task of partial label learning is to induce a multi-class classifier $g : \mathcal{X} \rightarrow \mathcal{Y}$ based on training set \mathcal{D} . Here the basic assumption under partial label learning framework is that the ground truth label y_i of x_i resides in S_i but it is not accessible to the algorithm during training phase.

The common strategy to train partial label predictive model is disambiguation, i.e. differentiating the modeling outputs of individual candidate labels so as to recover ground-truth labeling information. There are mainly two disambiguation strategies, i.e. the averaging-based strategy and the identification-based strategy. For averaging-based disambiguation, all the potential labels of each example are treated equally and the prediction is made by averaging

their modeling outputs [7, 13, 27]. For identification-based disambiguation, the ground-truth label is regarded as latent variable and identified through iterative refining procedure [5, 15, 16, 18].

Recently, feature-aware disambiguation method [28] was proposed to generate different labeling confidences over candidate label set by utilizing the graph structure of feature space. To facilitate the disambiguation process, feature-aware disambiguation makes use of the smoothness assumption that examples close to each other in the feature space will tend to share identical label in the label space. The approach proposed in [28] works in a two-stage manner which firstly generates latent labeling confidences over candidate label set and then learns a multi-class model by fitting a multi-output regressor with the generated labeling confidences.

Although the results in [28] have shown the effectiveness of feature-aware disambiguation for partial label learning, one potential drawback of this strategy lies in that real-world data always contain lots of noise and outliers that make the similarity graph obtained by original features less reliable. To this end, a novel approach for partial label learning is proposed in this paper based on *adaptive graph guided disambiguation* (PL-AGGD). Compared with fixed graph, adaptive graph could be more robust and accurate to reveal the intrinsic manifold structure within the data. Moreover, instead of the two-stage strategy in previous learning algorithms, our approach performs label disambiguation and predictive model training simultaneously. Specifically, we present a unified framework which enables the joint optimization of ground-truth labeling confidences, similarity graph and model parameters, which are solved based on an alternating procedure. Extensive experiments on controlled UCI data sets as well as real-world partial label data sets show that PL-AGGD performs favorably against other state-of-the-art partial label learning approaches.

The rest of the paper is organized as follows. We review related work in Section 2, and introduce the proposed approach in Section 3; Section 4 presents the settings and results of the experiments, followed by the conclusion in Section 5.

2 RELATED WORK

In partial label learning, the ground truth label of a training example resides in its candidate label set and thus is not accessible to the algorithm during training phase. Therefore, partial label learning can be regarded as a weakly-supervised learning framework with ambiguous labeling information.

To put the partial label learning problem into perspective, it is useful to lay out several related learning scenarios, including semi-supervised learning, multi-instance learning, multi-label learning and multi-instance multi-label learning. In semi-supervised learning [32], training examples are either explicitly labeled or unlabeled, while in partial label learning training examples are partially labeled. In multi-instance learning [1, 2], examples are not individually labeled but grouped into bags, while in partial label learning labels are assigned at the level of individual instances. In multi-label learning [10, 21, 29], each example is assigned with multiple labels all of which are valid, while in partial label learning the set of labels assigned to training examples are only candidate ones. Multi-instance multi-label learning [31] can be considered

as a generalized version of multi-instance learning and multi-label learning, where training instances are associated with not only multiple instances but also multiple labels. Partial label learning ignores any bag structure and views each instance independently, while each multi-instance multi-label example can be transformed into a number of partial label examples by treating the assigned class labels as candidate ones for each instance in the bag [16].

Existing partial label learning approaches try to solve the learning task by disambiguating the candidate label set, which can be achieved in two basic strategies. One intuitive strategy is to treat all the candidate labels of an example in an equal manner and the prediction is made by averaging their modeling outputs. For predictive models $g(\mathbf{x}, \mathbf{W})$ (\mathbf{W} denotes the model parameters), the averaged output over all candidate labels, i.e. $\frac{1}{|S_i|} \sum_{y \in S_i} g_y(\mathbf{x}_i, \mathbf{W})$, is distinguished from the outputs from non-candidate labels, i.e. $g_y(\mathbf{x}_i, \mathbf{W})$ ($y \notin S_i$) [7]. For non-parametric models, the predicted label for test example is determined by voting among the candidate labels of its neighboring examples [13, 27]. One potential drawback of the averaging-based disambiguation strategy lies in that the essential modeling output from ground-truth label might be overwhelmed by the distractive outputs from false positive labels.

Though the averaging-based strategy is intuitive and easy to be implemented, its effectiveness is largely affected by the false positive labels whose outputs would overwhelm the essential output yielded by the ground-truth label. Therefore, another strategy is to disambiguate the candidate label set by identifying the ground-truth label. Existing approaches following this strategy take the ground-truth label as latent variable which is identified as: $\hat{y}_i = \arg \max_{y \in S_i} g_y(\mathbf{x}_i, \mathbf{W})$. Generally, the latent variable and model parameters are refined iteratively via EM procedure which optimizes objective function defined according to the maximum likelihood criterion: $\sum_{i=1}^m \log(\sum_{y \in S_i} g_y(\mathbf{x}_i, \mathbf{W}))$ [15, 16], or the maximum margin criterion: $\sum_{i=1}^m (\max_{y \in S_i} g_y(\mathbf{x}_i, \mathbf{W}) - \max_{y \notin S_i} g_y(\mathbf{x}_i, \mathbf{W}))$ [18, 24]. One potential drawback of the identification-based disambiguation strategy lies in that, rather than recovering the ground-truth label, the identified label might turn out to be false positive label in the candidate label set.

The recently-proposed feature-aware disambiguation strategy [9, 28] makes use of the local manifold structure in feature space to help disambiguate the candidate label set, which differs from both averaging-based and identification-based disambiguation strategies. One potential drawback of this strategy lies in that real-world data are usually contaminated by significant noises and outliers which make the manifold structure recovered from original feature space less reliable [23, 26]. In the next section, we will introduce a novel approach which disambiguates the candidate labels adaptively. By adopting an alternating optimization procedure, our approach can iteratively refine the labeling confidences, similarity graph and model parameters in a joint manner.

3 THE PROPOSED APPROACH

In this section, we introduce the proposed approach PL-AGGD. We first present a unified framework which aims to perform disambiguation, adaptive graph optimization and model learning simultaneously. After that, an efficient optimization procedure is introduced to tackle the proposed framework.

3.1 Adaptive Graph Guided Disambiguation

We denote $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m]^\top \in \mathbb{R}^{m \times d}$ as the feature matrix and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_m]^\top \in \{0, 1\}^{m \times q}$ as the partial label matrix where $y_{ij} = 1$ means that the j -th label is one of the candidate labels of \mathbf{x}_i . Given the training set \mathcal{D} , a weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{S})$ is constructed over the training examples, where $\mathcal{V} = \{\mathbf{x}_i | 0 \leq i \leq m\}$ corresponds to the set of vertices and $\mathcal{E} = \{(\mathbf{x}_i, \mathbf{x}_j) | \mathbf{x}_i \in \text{KNN}(\mathbf{x}_j), i \neq j\}$ corresponds to the set of edges from \mathbf{x}_i to \mathbf{x}_j iff \mathbf{x}_i is among the k -nearest neighbors of \mathbf{x}_j . Furthermore, $\mathbf{S} \in \mathbb{R}^{m \times m}$ corresponds to the non-negative weight matrix where $s_{ij} = 0$ if $(\mathbf{x}_i, \mathbf{x}_j) \notin \mathcal{E}$.

For each training example (\mathbf{x}_i, S_i) , we aim to generate a normalized real-valued vector $\mathbf{f}_i \in \mathbb{R}^q$ where each f_{il} represents the labeling confidence of the l -th label being the ground-truth label for \mathbf{x}_i . The labeling confidence vector \mathbf{f}_i satisfies the following constraints: (i) $\sum_{l=1}^q f_{il} = 1$, (ii) $f_{il} \geq 0$ ($\forall y_{il} = 1$) and (iii) $f_{il} = 0$ ($\forall y_{il} = 0$). Here the second constraint implies that the latent ground-truth label resides in the candidate label set, and the third constraint guarantees that the labeling confidence of each non-candidate label must be 0. Once the normalized labeling confidence vectors have been generated, the original partial label training set \mathcal{D} can be transformed into its disambiguated counterpart: $\bar{\mathcal{D}} = \{(\mathbf{x}_i, \mathbf{f}_i) | 1 \leq i \leq m\}$. Then the predictive model can be induced by performing multi-output regression based on the disambiguation results $\bar{\mathcal{D}}$.

Feature-aware disambiguation differentiates the candidate label set by exploiting graph structure of the feature space. Specifically, the similarity graph weight matrix \mathbf{S} is constructed by solving the following linear least square problem:

$$\min_{\mathbf{S}} \sum_{j=1}^m \left\| \mathbf{x}_j - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{E}} s_{ij} \mathbf{x}_i \right\|_2^2 \quad (1)$$

$$\text{s.t. } \mathbf{S} \mathbf{1}_m = \mathbf{1}_m, s_{ij} \geq 0 (\forall (\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{E}), s_{ij} = 0 (\forall (\mathbf{x}_i, \mathbf{x}_j) \notin \mathcal{E})$$

where $\mathbf{1}_m$ is an all 1 vector with size m .

Then, by exploiting the smoothness assumption that the manifold structure in the feature space should also be preserved in the label space, the labeling confidence matrix $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_m]^\top \in \mathbb{R}^{m \times q}$ can be determined by solving the following problem:

$$\min_{\mathbf{F}} \sum_{j=1}^m \left\| \mathbf{f}_j - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{E}} s_{ij} \mathbf{f}_i \right\|_2^2 \quad (2)$$

$$\text{s.t. } \mathbf{F} \mathbf{1}_q = \mathbf{1}_m, f_{il} \geq 0 (\forall y_{il} = 1), f_{il} = 0 (\forall y_{il} = 0)$$

In previous feature-aware disambiguation, the problem (1) and (2) are solved independently, thus the disambiguation process is actually guided by a fixed graph. It is challenging for the fixed similarity graph to reveal the intrinsic structure within the data since noise and outliers widely exist, and thus we consider to get the similarity weight matrix in an adaptive way. We obtain the similarity weight matrix and simultaneously differentiate the labeling confidence to achieve better disambiguation results. By solving problems (1) and (2) jointly, the similarity graph weights are not only determined by the distance information of features, but also adjusted based on the feedbacks from label space. Compared to fixed graph, adaptive graph could be more robust and accurate.

Furthermore, instead of the two-stage (disambiguation and model induction) strategy in previous algorithms, we perform label disambiguation and predictive model training simultaneously. Thus the labeling confidence can be optimized by considering both the manifold assumption and model outputs. We present a unified framework to make labeling confidence, similarity weights and predictive model be jointly optimized to achieve strong generalization performance. We denote matrix $\mathbf{W} \in \mathbb{R}^{d \times q}$ as the model parameters, and use the least squares loss to optimize the predictive model:

$$\ell(g(\mathbf{x}_i, \mathbf{W}), \mathbf{f}_i) = \left\| g(\mathbf{x}_i, \mathbf{W}) - \mathbf{f}_i \right\|_2^2 \quad (3)$$

We adopt the widely-used squared Frobenius norm for the regularization term to control the model complexity. Then the objective function is shown as below:

$$\begin{aligned} \min_{\mathbf{S}, \mathbf{F}, \mathbf{W}} & \sum_{j=1}^m \left\| g(\mathbf{x}_j, \mathbf{W}) - \mathbf{f}_j \right\|_2^2 + \mu \sum_{j=1}^m \left\| \mathbf{f}_j - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{E}} s_{ij} \mathbf{f}_i \right\|_2^2 \\ & + \gamma \sum_{j=1}^m \left\| \mathbf{x}_j - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{E}} s_{ij} \mathbf{x}_i \right\|_2^2 + \lambda \|\mathbf{W}\|_{\text{F}}^2 \end{aligned} \quad (4)$$

$$\text{s.t. } \mathbf{S} \mathbf{1}_m = \mathbf{1}_m, \mathbf{0}_{m \times m} \leq \mathbf{S} \leq \mathbf{N}, \mathbf{F} \mathbf{1}_q = \mathbf{1}_m, \mathbf{0}_{m \times q} \leq \mathbf{F} \leq \mathbf{Y}$$

where $\mathbf{N} \in \{0, 1\}^{m \times m}$ is defined as: $n_{ij} = 1$ if $(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{E}$ and $n_{ij} = 0$ otherwise. $\mathbf{0}_{m \times m}$ is the $m \times m$ all 0 matrix. λ is the regularization parameter, and μ, γ are the trade-off parameters for reconstruction loss of label and feature space.

3.2 Alternative Optimization

Since problem (4) contains three sets of variables with different regularizations and constraints, it is hard to be tackled directly. Thus we apply the alternative optimization approach to solve this problem. Specifically, we iteratively optimize each set of variables by fixing other sets of variables until convergence or the maximum number of iterations is reached.

Update S With fixed \mathbf{F} and \mathbf{W} , the optimization problem (4) can be stated as follows:

$$\begin{aligned} \min_{\mathbf{S}} & \mu \sum_{j=1}^m \left\| \mathbf{f}_j - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{E}} s_{ij} \mathbf{f}_i \right\|_2^2 + \gamma \sum_{j=1}^m \left\| \mathbf{x}_j - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{E}} s_{ij} \mathbf{x}_i \right\|_2^2 \\ \text{s.t. } & \mathbf{S} \mathbf{1}_m = \mathbf{1}_m, \mathbf{0}_{m \times m} \leq \mathbf{S} \leq \mathbf{N} \end{aligned} \quad (5)$$

The similarity vector of each sample is independent, thus we fix other variables and optimize each similarity vector one by one. We consider to solve the following problem for the j -th sample in the rest of this subsection:

$$\begin{aligned} \min_{\mathbf{s}_j} & \mu \left\| \mathbf{f}_j - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{E}} s_{ij} \mathbf{f}_i \right\|_2^2 + \gamma \left\| \mathbf{x}_j - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{E}} s_{ij} \mathbf{x}_i \right\|_2^2 \\ \text{s.t. } & \mathbf{s}_j^\top \mathbf{1}_m = 1, \mathbf{0}_m \leq \mathbf{s}_j \leq \mathbf{n}_j \end{aligned} \quad (6)$$

Note that for \mathbf{s}_j , there are only k non-negative elements we need to update. Let $\hat{\mathbf{s}}_j \in \mathbb{R}^k$ denotes the similarity weight vector with each element representing a weight value that characterizes the relative importance of a neighboring sample in reconstructing \mathbf{x}_j . Denote matrix $\mathbf{D}^{fj} = [\mathbf{f}_j - \mathbf{f}_{N_j(1)}, \mathbf{f}_j - \mathbf{f}_{N_j(2)}, \dots, \mathbf{f}_j - \mathbf{f}_{N_j(k)}]^\top \in \mathbb{R}^{k \times q}$ and matrix $\mathbf{D}^{xj} = [\mathbf{x}_j - \mathbf{x}_{N_j(1)}, \mathbf{x}_j - \mathbf{x}_{N_j(2)}, \dots, \mathbf{x}_j - \mathbf{x}_{N_j(k)}]^\top \in \mathbb{R}^{k \times d}$.

Then the optimization problem (6) can be re-written as follows:

$$\begin{aligned} \min_{\hat{\mathbf{s}}_j} \hat{\mathbf{s}}_j^\top (\mu \mathbf{G}^{fj} + \gamma \mathbf{G}^{xj}) \hat{\mathbf{s}}_j \\ \text{s.t. } \hat{\mathbf{s}}_j^\top \mathbf{1}_k = 1, \mathbf{0}_k \leq \hat{\mathbf{s}}_j \leq \mathbf{1}_k \end{aligned} \quad (7)$$

where $\mathbf{G}^{fj} = \mathbf{D}^{fj}(\mathbf{D}^{fj})^\top \in \mathbb{R}^{k \times k}$ and $\mathbf{G}^{xj} = \mathbf{D}^{xj}(\mathbf{D}^{xj})^\top \in \mathbb{R}^{k \times k}$ are two Gram matrices corresponding to label space and feature space respectively. The optimization problem (7) is a standard Quadratic Programming (QP) problem with only k variables which can be efficiently solved by off-the-shelf QP tools. After each $\hat{\mathbf{s}}_i$ is solved, we concatenate them together and obtain the updated graph matrix \mathbf{S} .

Update F While \mathbf{S} and \mathbf{W} are fixed, the optimization problem (4) can be stated as follows:

$$\begin{aligned} \min_{\mathbf{F}} \|\mathbf{H} - \mathbf{F}\|_{\text{F}}^2 + \mu \sum_{j=1}^m \left\| \mathbf{f}_j - \sum_{(\mathbf{x}_i, \mathbf{x}_j) \in \mathcal{E}} s_{ij} \mathbf{f}_i \right\|_2^2 \\ \text{s.t. } \mathbf{F} \mathbf{1}_q = \mathbf{1}_m, \mathbf{0}_{m \times q} \leq \mathbf{F} \leq \mathbf{Y} \end{aligned} \quad (8)$$

where $\mathbf{F} = [\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_m]^\top \in \mathbb{R}^{m \times q}$ and $\mathbf{H} \in \mathbb{R}^{m \times q}$ is denoted as the modeling output matrix.

Let $\tilde{\mathbf{f}} = \text{vec}(\mathbf{F}) \in [0, 1]^{mq}$ where $\text{vec}(\bullet)$ is the vectorization operator. Likewise, $\tilde{\mathbf{h}} = \text{vec}(\mathbf{H}) \in \mathbb{R}^{mq}$ and $\tilde{\mathbf{y}} = \text{vec}(\mathbf{Y}) \in \{0, 1\}^{mq}$. We define a square matrix $\mathbf{T} = \mathbf{S}\mathbf{S}^\top + (\mathbf{S}^\top \mathbf{1}_{m \times m} \mathbf{S}) \odot \mathbf{I}_{m \times m} - 2\mathbf{W} + \frac{1}{\mu} \mathbf{I}_{m \times m}$ where \odot is the Hadamard product (the entrywise product) and $\mathbf{I}_{m \times m}$ is an identity matrix with m rows and m columns. Then the optimization problem (8) can be written as follows:

$$\begin{aligned} \min_{\tilde{\mathbf{f}}} \frac{1}{2} \tilde{\mathbf{f}}^\top (\mathbf{\Lambda} + \mathbf{\Lambda}^\top) \tilde{\mathbf{f}} - \frac{2}{\mu} \tilde{\mathbf{h}}^\top \tilde{\mathbf{f}} \\ \text{s.t. } \mathbf{0}_{mq} \leq \tilde{\mathbf{f}} \leq \tilde{\mathbf{y}}, \sum_{j=1, j \neq m}^{mq} \tilde{\mathbf{f}}_j = 1 \quad (\forall 0 \leq i \leq m-1) \end{aligned} \quad (9)$$

where $\mathbf{\Lambda} \in \mathbb{R}^{mq \times mq}$ is defined as follows:

$$\mathbf{\Lambda} = \begin{bmatrix} \mathbf{T} & \mathbf{0}_{m \times m} & \cdots & \mathbf{0}_{m \times m} \\ \mathbf{0}_{m \times m} & \mathbf{T} & \ddots & \vdots \\ \vdots & \ddots & \ddots & \mathbf{0}_{m \times m} \\ \mathbf{0}_{m \times m} & \cdots & \mathbf{0}_{m \times m} & \mathbf{T} \end{bmatrix} \quad (10)$$

The optimization problem (9) also corresponds a standard Quadratic Programming problem, which can be solved by off-the-shelf QP tools. The second constraint of problem (9) can be implemented in matrix form when using QP tools. Specifically, let $\mathbf{B} = [\mathbf{I}_{m \times m}, \mathbf{I}_{m \times m}, \dots, \mathbf{I}_{m \times m}] \in \{0, 1\}^{mq \times mq}$, then the constraint can be represented as $\mathbf{B}\tilde{\mathbf{f}} = \mathbf{1}_m$. Note that this QP problem have mq variables and $m(q+1)$ constraints, thus the computational complexity would be demanding when mq is large. Following [28] we employ alternating optimization strategy for large scale datasets. For labeling confidence vector \mathbf{f}_j , it can be optimized by fixing all other labeling confidence vectors:

$$\begin{aligned} \min_{\mathbf{f}_j} (t_{jj} + \frac{1}{\mu}) \mathbf{f}_j^\top \mathbf{f}_j + \left(\sum_{i=1, i \neq j}^m (t_{ij} + t_{ji}) \mathbf{f}_i^\top - \frac{2}{\mu} \mathbf{h}_i^\top \right) \mathbf{f}_j \\ \text{s.t. } \mathbf{0}_q \leq \mathbf{f}_j \leq \mathbf{y}_j, \mathbf{f}_j \mathbf{1}_q = 1 \end{aligned} \quad (11)$$

Now the problem (9) is transformed into a series of QP subproblems with q variables and $q+1$ constraints which can be solved more efficiently.

Table 1: The pseudo-code of PL-AGGD

Input:

- \mathcal{D} : the partially labeled training set $\{(\mathbf{x}_i, S_i) | 1 \leq i \leq m\}$
- k : the number of nearest neighbors
- μ, γ, λ : the trade-off parameters
- \mathbf{x} : the unseen instance

Output:

- y : the predicted label for instance \mathbf{x}

Process:

- 1: Calculate the kernel matrix $\mathbf{K} = [\kappa(\mathbf{x}_i, \mathbf{x}_j)]_{m \times m}$ ($1 \leq i, j \leq m$);
- 2: Set the similarity graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{S})$;
- 3: Initialize \mathbf{S} by solving the problem (1);
- 4: Initialize \mathbf{F} by solving the problem (2);
- 5: **repeat**
- 6: Update \mathbf{A} according to (17);
- 7: Update model outputs $\mathbf{H} = \frac{1}{2\lambda} \mathbf{K}\mathbf{A}$;
- 8: **for** $j = 1$ **to** m **do**
- 9: Update the j -th column of \mathbf{S} by solving (7);
- 10: **end for**
- 11: Update \mathbf{F} by solving (9) or iteratively solving a series subproblems shown as (11);
- 12: **until** convergence or the maximum number of iterations
- 13: **return** y according to (18).

Update W With the simple linear model $g(\mathbf{x}_i, \mathbf{W}) = \mathbf{W}^\top \mathbf{x}_i$ while fixing \mathbf{F} and \mathbf{S} , the problem (4) can be stated as follows:

$$\min_{\mathbf{W}} \sum_i \|\mathbf{W}^\top \mathbf{x}_i - \mathbf{f}_i\|_2^2 + \lambda \|\mathbf{W}\|_{\text{F}}^2 \quad (12)$$

This regularized least squares problem is simple and can be easily solved by gradient descent method.

To achieve better performance of the predictive model, we can further facilitate a kernel extension for the general nonlinear case. Let $\phi(\bullet) : \mathbb{R}^d \rightarrow \mathbb{R}^h$ denotes the feature mapping that maps the feature space to some higher dimensional Hilbert space with h dimensions. The predictive function with kernel extension can be represented as $g(\mathbf{x}_i, \mathbf{W}) = \mathbf{W}^\top \phi(\mathbf{x}_i)$. Then we have the following problem:

$$\min_{\mathbf{W}} \sum_i \|\epsilon_i\|_2^2 + \lambda \|\mathbf{W}\|_{\text{F}}^2 \quad \text{s.t. } \mathbf{f}_i = \mathbf{W}^\top \phi(\mathbf{x}_i) - \epsilon_i \quad (13)$$

By defining a matrix $\mathbf{E} = [\epsilon_1, \epsilon_2, \dots, \epsilon_3]^\top \in \mathbb{R}^{m \times q}$, the above problem can be re-written as the following matrix form:

$$\min_{\mathbf{W}} \text{tr}(\mathbf{E}^\top \mathbf{E}) + \lambda \text{tr}(\mathbf{W}^\top \mathbf{W}) \quad \text{s.t. } \mathbf{F} = \mathbf{\Phi}\mathbf{W} - \mathbf{E} \quad (14)$$

where $\mathbf{\Phi} = [\phi(\mathbf{x}_1), \phi(\mathbf{x}_2), \dots, \phi(\mathbf{x}_m)]^\top$. The lagrangian function of this problem is:

$$\mathcal{L}(\mathbf{W}, \mathbf{E}, \mathbf{A}) = \text{tr}(\mathbf{E}^\top \mathbf{E}) + \lambda \text{tr}(\mathbf{W}^\top \mathbf{W}) - \text{tr}(\mathbf{A}^\top (\mathbf{\Phi}\mathbf{W} - \mathbf{E} - \mathbf{F})) \quad (15)$$

where $\mathbf{A} \in \mathbb{R}^{m \times q}$ stores the Lagrange multipliers. We know that the optimum solution of (15) need to satisfy the following conditions:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{W}} &= 2\lambda \mathbf{W} - \Phi^\top \mathbf{A} = \mathbf{0} & \frac{\partial \mathcal{L}}{\partial \mathbf{E}} &= 2\mathbf{E} + \mathbf{A} = \mathbf{0} \\ \frac{\partial \mathcal{L}}{\partial \Psi} &= \Phi \mathbf{W} - \mathbf{E} - \mathbf{F} = \mathbf{0} \end{aligned} \quad (16)$$

From these linear equations of we can obtain:

$$\mathbf{W} = \frac{\Phi^\top \mathbf{A}}{2\lambda} \quad \mathbf{A} = \left(\frac{1}{2\lambda} \mathbf{K} + \frac{1}{2} \mathbf{I}_{m \times m} \right)^{-1} \mathbf{F} \quad (17)$$

where $\mathbf{K} = \Phi \Phi^\top$ with its element $k_{ij} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$ based on the corresponding kernel function $\kappa(\cdot, \cdot)$. For PL-AGGD, Gaussian kernel $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{\sigma^2})$ is employed with σ being the average distance among each pair of training examples. Then, by incorporating the specified kernel function, the modeling output matrix is denoted by $\mathbf{H} = \Phi \mathbf{W} = \frac{1}{2\lambda} \mathbf{K} \mathbf{A}$. Furthermore, the predicted label of the test example \mathbf{x} yielded by our approach is given as:

$$\mathbf{y}^* = \arg \min_k \sum_{i=1}^m a_{ik} \kappa(\mathbf{x}, \mathbf{x}_i) \quad (18)$$

The pseudo code of PL-AGGD is summarized in Table 1. We can observe that the first process of our approach is to initialize \mathbf{S} and \mathbf{F} (Steps 2-4) by adopting feature-aware disambiguation. After that, the alternative optimization strategy is adopted to learn an adaptive graph to guide label disambiguation (Steps 5-12) and train the predictive model simultaneously. Finally, the unseen instance is classified based on the learned model (Step 13).

4 EXPERIMENTS

4.1 Experimental Setup

In this section, we conduct two series of comparative experiments based on controlled UCI data sets as well as real-world partial label data sets. Characteristics of the controlled UCI data sets and real-world data sets are summarized in Table 2 and Table 5 respectively.

Following the widely-used controlling protocol in previous partial label learning research [5, 7, 16], an artificial partial label data set can be generated from a multi-class UCI data set with controlling parameters p , r and ϵ . Here, p controls the proportion of examples which are partially labeled (we simply set $p = 1$ across all our experiments), r controls the number of false positive labels in the candidate label set (i.e. $|S_i| = r + 1$), and in our setting the ambiguous labels in the training set are generated without replacement. ϵ is the ambiguity degree which controls the co-occurring probability between one coupling candidate label and the ground-truth label. This is achieved by first choosing at random for each label a dominant co-occurring label which is sampled with probability ϵ , and then the rest of the labels are sampled uniformly with probability $(1 - \epsilon)/(L - 2)$ (there is a single extra label per example).

In addition to artificial data sets, a number of real-world partially labeled data sets have been collected from several task domains (see in Table 5) including Lost [6], Soccer Player [25], Yahoo! News [12] from automatic face naming, MSRCv2 [16] from object classification, FG-NET [19] from facial age estimation, and

Table 2: Characteristics of the UCI data sets.

| Data set | #examples | #features | #classes |
|-----------|-----------|-----------|----------|
| vehicle | 846 | 18 | 4 |
| sensor | 5456 | 24 | 4 |
| steel | 1941 | 27 | 7 |
| segment | 2310 | 18 | 7 |
| ecoli | 336 | 7 | 8 |
| winerate | 1599 | 11 | 10 |
| pendigits | 10992 | 16 | 10 |
| abalone | 4177 | 7 | 29 |

BirdSong [3] from bird song classification. For the task of automatic face naming, faces cropped from an image or video are represented as instances while names extracted from the associated captions or subtitles are regarded as candidate labels. Specifically, top frequent names are retained for the Labeled Yahoo! News data set. For the task of object classification, image segmentations are represented as instances while objects appearing within the same image are regarded as candidate labels. For the task of facial age estimation, human faces with landmarks are represented as instances while ages annotated by ten crowdsourced labelers together with the ground-truth age are regarded as candidate labels. For the task of bird song classification, singing syllables of the birds are represented as instances while bird species jointly singing during a 10-seconds period are regarded as candidate labels. The average number of candidate labels (avg. #CLs) for each real-world partial label data set is also recorded in Table 5.

Five state-of-the-art partial label learning algorithms are employed for comparative studies, each implemented with hyper parameters setup suggested in respective literatures:

- PL-KNN [13]: A k -nearest neighbor approach to partial label learning which conducts averaging-based disambiguation [suggested setup: $k = 10$];
- CLPL [7]: An averaging-based disambiguation approach based on a convex learning formulation which minimizes a convex loss function under the partial label setting [suggested setup: SVM with squared hinge loss];
- PL-SVM [18]: A maximum margin approach to partial label learning which conducts identification-based disambiguation [suggested setup: regularization parameter pool with $\{0.01, 0.1, 1, 10, 100\}$];
- LSB-CMM [16]: A maximum likelihood approach to partial label learning which conducts identification-based disambiguation [suggested setup: mixture components = q].
- PL-LEAF [28]: Two-stage approach which learns from partial label examples based on feature-aware disambiguation [$k = 10$, $C_1 = 10$ and $C_2 = 1$].

For PL-AGGD, the parameters employed are set as $k = 10$, $T = 20$, $\lambda = 1$, $\mu = 1$ and $\gamma = 0.05$. The sensitivity analysis of parameter configuration is conducted in Subsection 4.3. We perform ten runs of 50%/50% random train/test splits on each artificial as well as real-world partial label data set, and the mean accuracies (with standard deviations) are recorded for all algorithms.

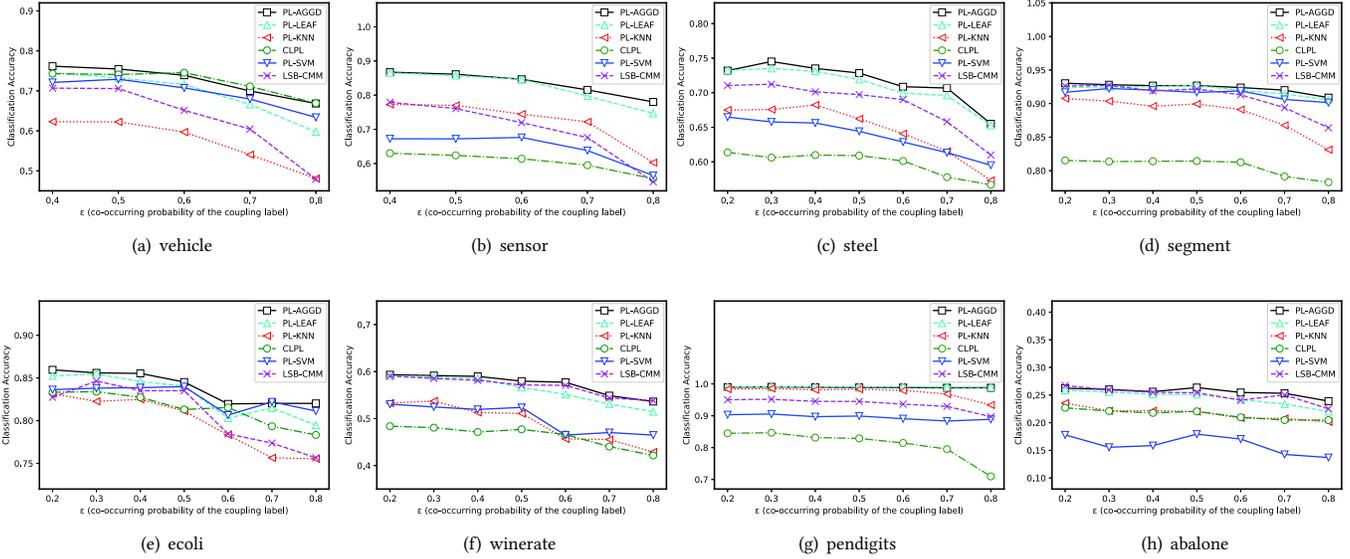


Figure 1: Classification accuracy of each comparing algorithm with varying ϵ (co-occurring probability of the coupling label) and one false positive candidate label ($r = 1$).

4.2 Experimental Results

Controlled UCI Data Sets Figure 1 illustrates the classification accuracy of each comparing algorithm on the controlled UCI data sets as the co-occurring probability ϵ varies from 0.1 to 0.8 with step size 0.1. For any ground-truth label $y \in Y$, one extra label $y' \neq y$ is designated as the coupling label which co-occurs with y in the candidate label set with probability ϵ ($r = 1$).

In Table 3, we vary the ambiguity size: the number of extra labels associated with each example. For each example, along with the ground-truth label, r extra class labels will be randomly picked up to constitute the candidate label set.

As shown in Figure 1 and Table 3, PL-AGGD achieves superior or competitive performance against the comparing algorithms. Out of the 75 statistical tests on all 8 UCI data sets (Table 4 with 52 configurations from Figure 1 and 23 configurations from Table 3), it is shown that:

- Compared to the averaging-based disambiguation approaches, PL-AGGD achieves superior performance against PL-KNN and CLPL in most cases.
- Compared to the identification-based disambiguation approaches, PL-AGGD achieves superior performance against PL-SVM and LSB-CMM in 86.7% and 72% cases respectively. Furthermore, the performance of PL-AGGD is inferior to LSB-CMM in only 1 case and has been outperformed by PL-SVM in only 4 cases.
- Compared to existing feature-aware approach PL-LEAF, PL-AGGD achieves superior performance in 56% cases and outperformed by PL-LEAF in only 5 cases. Furthermore, from Figure 1 and Table 3 we can observe that PL-AGGD is more robust when class ambiguity degree ϵ or ambiguity size r is large.

Real-World Data Sets Table 6 reports the classification accuracy of each comparing algorithm on the real-world partial label data sets.^a Note that the average number of candidate labels (avg. #CLs) of the dataset FG-NET is quite large, which causes an extremely low classification accuracy of each algorithm based on training examples with partial labels. The common evaluation criteria on this task is mean average error (MAE) between the predicted age and the ground-truth age. In Table 6, for better evaluation of this facial age estimation task, two extra experiments are conducted on the dataset FG-NET where a test example is considered to be correctly classified if the difference between the predicted age and the ground-truth age is no more than 3 years (MAE3) or 5 years (MAE5).

As shown in Table 6, it is obvious that PL-AGGD significantly outperforms all the counterpart algorithms on these real-world datasets except for PL-SVM on Yahoo! News and PL-LEAF on MSRCv2. Furthermore, our approach is never outperformed by any comparing algorithm.

4.3 Further Analysis

Transductive Accuracy In addition to Table 6 on the inductive performance, it is also interesting to study the transductive performance of each comparing algorithm on classifying training examples [7, 28]. Transductive performance of the partial label learning algorithm reflects its ability in disambiguating the candidate label set. For PL-AGGD and PL-LEAF, the generated labeling confidence vector f_i can be used to predict the ground-truth label of training example x_i as $\hat{y}_i = \arg \max_{y_k \in S_i} f_{ik}$. For other approach, the ground-truth label is predicted by consulting the candidate label

^aData sets available at: http://palm.seu.edu.cn/zhangml/Resources.htm#partial_data

Table 3: Classification accuracy (mean±std) of each comparing algorithm on the controlled UCI data sets with the varying r (false positive candidate label). In addition, ●/○ indicates whether the performance of PL-AGGD is statistically superior/inferior to the comparing algorithm on each data set (pairwise t-test at 0.05 significance level).

| | r | PL-AGGD | PL-KNN | CLPL | PL-SVM | LSB-CMM | PL-LEAF |
|-----------|-----|-------------|--------------|--------------|--------------|--------------|--------------|
| vehicle | 1 | 0.770±0.025 | 0.641±0.017● | 0.756±0.025● | 0.727±0.027● | 0.720±0.022● | 0.753±0.024● |
| | 2 | 0.686±0.028 | 0.512±0.018● | 0.708±0.015● | 0.649±0.032● | 0.604±0.025● | 0.670±0.033● |
| sensor | 1 | 0.866±0.004 | 0.771±0.006● | 0.633±0.010● | 0.682±0.010● | 0.775±0.010● | 0.865±0.005 |
| | 2 | 0.811±0.008 | 0.692±0.015● | 0.605±0.013● | 0.650±0.023● | 0.649±0.006● | 0.788±0.006● |
| segment | 1 | 0.927±0.006 | 0.907±0.009● | 0.811±0.008● | 0.915±0.010● | 0.925±0.008 | 0.929±0.005 |
| | 3 | 0.908±0.007 | 0.878±0.000● | 0.805±0.016● | 0.916±0.004○ | 0.896±0.011● | 0.912±0.010○ |
| | 5 | 0.862±0.019 | 0.512±0.032● | 0.780±0.016● | 0.816±0.066● | 0.765±0.026● | 0.847±0.020● |
| steel | 1 | 0.733±0.019 | 0.677±0.009● | 0.614±0.016● | 0.660±0.013● | 0.708±0.013● | 0.730±0.017 |
| | 3 | 0.703±0.014 | 0.605±0.011● | 0.579±0.026● | 0.612±0.026● | 0.656±0.009● | 0.693±0.016● |
| | 5 | 0.580±0.028 | 0.329±0.024● | 0.501±0.038● | 0.474±0.056● | 0.528±0.029● | 0.550±0.034● |
| ecoli | 1 | 0.837±0.024 | 0.804±0.033● | 0.815±0.026● | 0.828±0.033 | 0.825±0.029 | 0.833±0.024 |
| | 3 | 0.828±0.028 | 0.796±0.027● | 0.805±0.034● | 0.816±0.026 | 0.767±0.027● | 0.833±0.031 |
| | 5 | 0.727±0.063 | 0.633±0.068● | 0.697±0.065● | 0.667±0.086● | 0.622±0.070● | 0.716±0.067● |
| winerate | 1 | 0.588±0.014 | 0.533±0.018● | 0.469±0.016● | 0.512±0.018● | 0.581±0.012 | 0.587±0.012 |
| | 3 | 0.574±0.018 | 0.452±0.029● | 0.438±0.028● | 0.502±0.021● | 0.581±0.015○ | 0.549±0.018● |
| | 5 | 0.515±0.032 | 0.315±0.024● | 0.377±0.022● | 0.423±0.042● | 0.547±0.018○ | 0.469±0.037● |
| pendigits | 1 | 0.990±0.001 | 0.984±0.001● | 0.849±0.002● | 0.905±0.004● | 0.951±0.003● | 0.991±0.000○ |
| | 3 | 0.990±0.001 | 0.983±0.001● | 0.832±0.005● | 0.889±0.007● | 0.919±0.009● | 0.990±0.001 |
| | 5 | 0.989±0.001 | 0.966±0.002● | 0.830±0.006● | 0.857±0.055● | 0.882±0.004● | 0.988±0.001 |
| abalone | 1 | 0.260±0.008 | 0.230±0.006● | 0.227±0.008● | 0.158±0.020● | 0.265±0.014○ | 0.259±0.009 |
| | 4 | 0.253±0.005 | 0.187±0.004● | 0.219±0.010● | 0.149±0.030● | 0.256±0.006 | 0.248±0.005● |
| | 7 | 0.248±0.010 | 0.150±0.011● | 0.204±0.006● | 0.156±0.033● | 0.239±0.009● | 0.238±0.008● |
| | 10 | 0.241±0.009 | 0.122±0.008● | 0.190±0.007● | 0.184±0.019● | 0.230±0.012● | 0.227±0.008● |

Table 4: Win/tie/loss counts (pairwise t-test at 0.05 significance level) on the classification performance of PL-AGGD against each comparing algorithm.

| | PL-AGGD against | | | | |
|-----------------|-----------------|---------------|---------------|----------------|----------------|
| | PL-KNN | CLPL | PL-SVM | LSB-CMM | PL-LEAF |
| [Figure 1] | 52/0/0 | 47/5/0 | 45/7/0 | 38/13/1 | 29/20/3 |
| [Table 3] | 23/0/0 | 23/0/0 | 10/2/1 | 16/4/3 | 13/8/2 |
| In total | 75/0/0 | 70/5/0 | 65/9/1 | 54/17/4 | 42/28/5 |

set S_i , i.e. predicting $\hat{y}_i \in S_i$ with largest modeling output. Accordingly, Table 7 reports the transductive accuracy of each comparing algorithm together with the outcomes of pairwise t-tests at 0.05 significance level. Out of the 40 statistical tests (8 data sets \times 5 comparing algorithm), it is shown that:

- Compared to averaging-based disambiguation approaches, PL-AGGD is outperformed by PL-KNN on the FG-NET data set and outperformed by CLPL on the Lost and FG-NET data sets. In terms of the rest statistical tests, the performance of

PL-AGGD is superior or at least comparable to PL-KNN and CLPL.

- Compared to identification-based disambiguation approaches, PL-AGGD is outperformed by PL-SVM on the Yahoo! News data set. In terms of the rest statistical tests, the performance of PL-AGGD is superior or at least comparable to PL-SVM and LSB-CMM.
- Compared to existing feature-aware approach, the disambiguation performance of PL-AGGD is superior or at least comparable to PL-LEAF on all data sets.

Parameter Sensitivity We also study the sensitivity of PL-AGGD with respect to its three parameters λ , μ and γ . Figure 2 shows the performance of PL-AGGD under different parameter configurations on Lost and BirdSong. As shown in Figure 2(a) and 2(b), when μ is small (it means that the manifold information of label space is hardly exploited when updating F), PL-AGGD gives a poor performance. As μ increases, PL-AGGD starts to take into consideration the manifold information and the classification accuracy increases. However, if μ is too large, the classification accuracy will drop dramatically. For λ , the performance would also be at a low level when it is too small or too large. In practice, we suggest users to choose

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

| Data set | #examples | #features | #classes | avg. #CLs | Task Domain |
|---------------|-----------|-----------|----------|-----------|-------------------------------------|
| Lost | 1122 | 108 | 16 | 2.23 | <i>automatic face naming</i> [6] |
| MSRCv2 | 1758 | 48 | 23 | 3.16 | <i>object classification</i> [16] |
| FG-NET | 1002 | 262 | 78 | 7.48 | <i>facial age estimation</i> [19] |
| Birdsong | 4998 | 38 | 13 | 2.18 | <i>bird song classification</i> [3] |
| Soccer Player | 17472 | 279 | 171 | 2.09 | <i>automatic face naming</i> [25] |
| Yahoo! News | 22991 | 163 | 219 | 1.91 | <i>automatic face naming</i> [12] |

Table 6: Classification accuracy (mean±std) of each comparing algorithm on the real-world partial label data sets. In addition, ●/○ indicates whether the performance of PL-AGGD is statistically superior/inferior to the comparing algorithm on each data set (pairwise t-test at 0.05 significance level).

| | PL-AGGD | PL-KNN | CLPL | PL-SVM | LSB-CMM | PL-LEAF |
|---------------|-------------|--------------|--------------|--------------|--------------|--------------|
| Lost | 0.711±0.010 | 0.308±0.025● | 0.662±0.022● | 0.682±0.031● | 0.594±0.021● | 0.678±0.016● |
| MSRCv2 | 0.464±0.014 | 0.403±0.017● | 0.399±0.016● | 0.430±0.014● | 0.404±0.016● | 0.470±0.018 |
| FG-NET | 0.076±0.005 | 0.038±0.009● | 0.053±0.012● | 0.060±0.013● | 0.053±0.007● | 0.067±0.011● |
| FG-NET(MAE3) | 0.427±0.012 | 0.278±0.019● | 0.320±0.038● | 0.347±0.018● | 0.353±0.016● | 0.407±0.022● |
| FG-NET(MAE5) | 0.566±0.018 | 0.425±0.019● | 0.451±0.043● | 0.483±0.019● | 0.502±0.021● | 0.544±0.028● |
| BirdSong | 0.717±0.008 | 0.627±0.006● | 0.624±0.009● | 0.663±0.018● | 0.683±0.003● | 0.707±0.008● |
| Soccer Player | 0.524±0.005 | 0.494±0.004● | 0.348±0.005● | 0.485±0.004● | 0.504±0.002● | 0.516±0.003● |
| Yahoo! News | 0.613±0.005 | 0.448±0.005● | 0.459±0.005● | 0.609±0.006 | 0.585±0.005● | 0.607±0.002● |

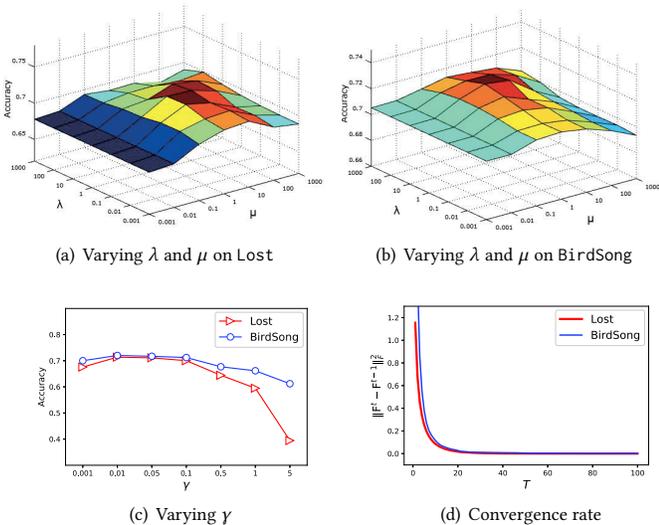


Figure 2: Parameter sensitivity analysis for PL-AGGD. (a) Classification accuracy of PL-AGGD on Lost by varying λ and μ ; (b) Classification accuracy of PL-AGGD on BirdSong by varying λ and μ ; (c) Classification accuracy of PL-AGGD on Lost and BirdSong by varying γ ; (d) Convergence curves of the labeling scores on Lost and BirdSong.

λ and μ around 1 for reliable usage. Another trade-off parameter γ

aims to control the model complexity. The classification accuracy curve of varying γ (Figure 2(c)) obviously accords with the fact that it is important to balance between overfitting and underfitting. In practice, it can be manually searched prior to training.

Figure 2(d) illustrates the convergence curve of PL-AGGD by using difference between the labeling confidence matrices (measured by L_2 norm $\|F^{(t)} - F^{(t-1)}\|_F^2$) of two adjacent iterations. We can see that the labeling confidence scores converge with increasing number of iterations (F becomes convergent when T reaches 20).

5 CONCLUSION

In this paper, the problem of partial label learning is studied and a novel approach based on adaptive graph guided disambiguation is proposed. Instead of employing a fixed similarity graph as previous feature-aware disambiguation strategy, our proposed approach PL-AGGD aims to learn from partial label examples by performing label disambiguation and predictive model training simultaneously. Extensive comparative studies clearly validate that adaptive graph guided disambiguation could be more robust and accurate to reveal the intrinsic manifold structure within the data.

The proposed approach PL-AGGD performs manifold structure discovery on original feature space, it would be interesting to integrate the feature embedding technique into PL-AGGD to achieve better performance. Besides, it would also be valuable to investigate the idea of adaptive graph to other weakly supervised learning setting, such as semi-supervised learning, incomplete multi-label learning, etc.

Table 7: Transductive accuracy (mean±std) of each comparing algorithm on the real-world partial label data sets. In addition, ●/○ indicates whether the performance of PL-AGGD is statistically superior/inferior to the comparing algorithm on each data set (pairwise t-test at 0.05 significance level).

| | PL-AGGD | PL-KNN | CLPL | PL-SVM | LSB-CMM | PL-LEAF |
|---------------|-------------|--------------|--------------|--------------|--------------|--------------|
| Lost | 0.830±0.015 | 0.586±0.024● | 0.848±0.019○ | 0.832±0.027 | 0.748±0.023● | 0.776±0.019● |
| MSRCv2 | 0.630±0.009 | 0.567±0.020● | 0.604±0.025● | 0.619±0.026● | 0.592±0.023● | 0.628±0.008 |
| FG-NET | 0.142±0.007 | 0.171±0.021○ | 0.160±0.010○ | 0.155±0.010 | 0.128±0.010● | 0.133±0.008● |
| FG-NET(MAE3) | 0.565±0.013 | 0.545±0.023● | 0.547±0.014● | 0.531±0.029● | 0.530±0.015● | 0.546±0.015● |
| FG-NET(MAE5) | 0.701±0.014 | 0.702±0.022 | 0.694±0.021 | 0.670±0.027● | 0.674±0.013● | 0.681±0.019● |
| BirdSong | 0.826±0.012 | 0.751±0.014● | 0.758±0.015● | 0.830±0.014 | 0.812±0.015 | 0.782±0.016● |
| Soccer Player | 0.715±0.005 | 0.652±0.004● | 0.626±0.003● | 0.710±0.006 | 0.684±0.003● | 0.701±0.004● |
| Yahoo! News | 0.837±0.003 | 0.700±0.003● | 0.746±0.008● | 0.842±0.002○ | 0.830±0.003● | 0.820±0.003● |

REFERENCES

- [1] Jaume Amores. 2013. Multiple instance classification: Review, taxonomy and comparative study. *Artificial intelligence* 201 (2013), 81–105.
- [2] Stuart Andrews, Ioannis Tsochantaridis, and Thomas Hofmann. 2003. Support vector machines for multiple-instance learning. In *Advances in Neural Information Processing Systems 15*, S. Becker, S. Thrun, and K. Obermayer (Eds.). MIT Press, Cambridge, MA, 577–584.
- [3] Forrest Briggs, Xiaoli Z Fern, and Raviv 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] Ching-Hui Chen, Vishal M Patel, and Rama Chellappa. 2018. Learning from ambiguously labeled face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 40, 7 (2018), 1653–1667.
- [5] Yi-Chen Chen, Vishal M Patel, Rama Chellappa, and P Jonathon Phillips. 2014. Ambiguously labeled learning using dictionaries. *IEEE Transactions on Information Forensics and Security* 9, 12 (2014), 2076–2088.
- [6] Timothee Cour, Benjamin Sapp, Chris Jordan, and Ben Taskar. 2009. Learning from ambiguously labeled images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Miami, FL, 919–926.
- [7] Timothee Cour, Ben Sapp, and Ben Taskar. 2011. Learning from partial labels. *Journal of Machine Learning Research* 12, May (2011), 1501–1536.
- [8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Miami, FL, 248–255.
- [9] Lei Feng and Bo 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.
- [10] Eva Gibaja and Sebastián Ventura. 2015. A tutorial on multilabel learning. *Comput. Surveys* 47, 3 (2015), 52.
- [11] Chen Gong, Tongliang Liu, Yuanyan Tang, Jian Yang, Jie Yang, and Dacheng Tao. 2018. A regularization approach for instance-based superset label learning. *IEEE Transactions on Cybernetics* 48, 3 (2018), 967–978.
- [12] Matthieu Guillaumin, Jakob Verbeek, and Cordelia 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.
- [13] Eyke Hüllermeier and Jürgen Beringer. 2006. Learning from ambiguously labeled examples. *Intelligent Data Analysis* 10, 5 (2006), 419–439.
- [14] Takashi Ishida, Gang Niu, Weihua Hu, and Masashi Sugiyama. 2017. Learning from complementary labels. In *Advances in Neural Information Processing Systems 30*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.). Curran Associates, Inc., 5639–5649.
- [15] Rong Jin and Zoubin 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.
- [16] Liping Liu and Thomas G 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.
- [17] Jie Luo and Francesco 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.
- [18] Nam Nguyen and Rich 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.
- [19] Gabriel Panis, Andreas Lanitis, Nicholas Tsapatsoulis, and Timothy F Cootes. 2016. Overview of research on facial ageing using the FG-NET ageing database. *IET Biometrics* 5, 2 (2016), 37–46.
- [20] George Papandreou, Liang-Chieh Chen, Kevin P Murphy, and Alan L Yuille. 2015. Weakly- and semi-supervised learning of a deep convolutional network for semantic image segmentation. In *Proceedings of the IEEE International Conference on Computer Vision*. Santiago, Chile, 1742–1750.
- [21] Grigorios Tsoumakas, Ioannis Katakis, and Ioannis Vlahavas. 2010. Mining multi-label data. In *Data Mining and Knowledge Discovery Handbook*. Springer, Berlin, 667–686.
- [22] Jing Wang and Min-Ling Zhang. 2018. Towards mitigating the class-imbalance problem for partial label learning. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. London, UK, 2427–2436.
- [23] Lichen Wang, Zhengming Ding, and Yun Fu. 2018. Adaptive graph guided embedding for multi-label annotation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*. Stockholm, Sweden, 2798–2804.
- [24] Fei Yu and Min-Ling Zhang. 2017. Maximum margin partial label learning. *Machine Learning* 106, 4 (2017), 573–593.
- [25] Zinan Zeng, Shijie Xiao, Kui Jia, Tsung-Han Chan, Shenghua Gao, Dong Xu, and Yi Ma. 2013. Learning by associating ambiguously labeled images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. Portland, OR, 708–715.
- [26] Lefei Zhang, Qian Zhang, Bo Du, Jane You, and Dacheng Tao. 2017. Adaptive manifold regularized matrix factorization for data clustering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. Melbourne, Australia, 3399–3405.
- [27] Min-Ling Zhang and Fei 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.
- [28] Min-Ling Zhang, Bin-Bin Zhou, and Xu-Ying Liu. 2016. 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, 1335–1344.
- [29] Min-Ling Zhang and Zhi-Hua Zhou. 2014. A review on multi-label learning algorithms. *IEEE Transactions on Knowledge and Data Engineering* 26, 8 (2014), 1819–1837.
- [30] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett (Eds.). Curran Associates, Inc., 649–657.
- [31] Zhi-Hua Zhou, Min-Ling Zhang, Sheng-Jun Huang, and Yu-Feng Li. 2012. Multi-instance multi-label learning. *Artificial Intelligence* 176, 1 (2012), 2291–2320.
- [32] Xiaojin Zhu and Andrew 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.