Partial Label Learning with Unlabeled Data

Qian-Wei Wang , Yu-Feng Li , Zhi-Hua Zhou
National Key Laboratory for Novel Software Technology
Nanjing University, Nanjing 210023, China
{wangqw, liyf, zhouzh}@lamda.nju.edu.cn

Abstract

Partial label learning deals with training examples each associated with a set of candidate labels, among which only one label is valid. Previous studies typically assume that the candidate label sets are provided for all training examples. In many real-world applications such as video character classification, however, it is generally difficult to label a large number of instances and there exists much data left to be unlabeled. We call this kind of problem *semi-supervised partial label learning*. In this paper, we propose the SSPL method to address this problem. Specifically, an iterative label propagation procedure between partial label examples and unlabeled instances is employed to disambiguate the candidate label sets of partial label examples as well as assign valid labels to unlabeled instances. The importance of unlabeled instances increases adaptively as the number of iteration increases, since they carry richer labeling information. Finally, unseen instances are classified based on the minimum reconstruction error on both partial labeled and unlabeled instances. Experiments on real-world data sets clearly validate the effectiveness of the proposed SSPL method.

1 Introduction

Conventional supervised learning often assumes that each training instance is associated with a ground-truth label. However, in many real-world applications, one can only get access to a candidate label set associated with each training instance among which only one label is valid. For example, an episode of a video or TV serials may contain several characters and their faces may appear simultaneously in a screenshot. We have scripts and dialogues or subtitles which indicate that who is in the given screenshot. So that we can ambiguously name a face appeared in a screenshot by a candidate label set which contains the names appeared in the scripts and dialogues corresponding to it, but can not tell which face is associated with which name. In order to deal with this kind of training examples, partial label learning (PLL) has recently been proposed [Cour *et al.*, 2011; Chen *et al.*, 2014; Yu and Zhang, 2017] and attracted considerable attention, which has consequently resulted in a large number of PLL methods [Hüllermeier and Beringer, 2006; Nguyen and Caruana, 2008; Liu and Dietterich, 2012; Zhang and Yu, 2015; Tang and Zhang, 2017].

In the previous studies on PLL, a basic assumption for training data is that all the candidate label sets are provided. However, in many real-world applications, such assumption is difficult to hold. Taking the above example again, although we can ambiguously label a face by a candidate label set which contains the names appeared in the scripts and dialogues, there still exist many screenshots that have no corresponding dialogues so that we have actually no label information for them. Similar situation occurs in many popular real-world applications such as web mining [Jie and Orabona, 2010], multimedia contents analysis [Zeng *et al.*, 2013], ecoinformatics [Liu and Dietterich, 2012], etc.

It is evident that neither PLL nor semi-supervised learning (SSL) can tackle the problem concerned in this paper. For example, PLL ignores the use of large amount of unlabeled instances that could be very useful; while SSL assumes that the ground-truth single-label is accessible to each labeled training example, which is not the case in our situation. Note that the data scenario studies in the paper are quite different from previous work. We call this kind of problem *semi-supervised partial label learning*.

In this paper, a novel algorithm named SSPL (Semi-Supervised Partial Label Learning), is proposed. It is crucial to disambiguate the candidate label sets of partial label examples at the same time as utilizing the data distribution information of unlabeled instances. In our method, an iterative label propagation procedure between partial labeled and unlabeled instances is employed to disambiguate the candidate label sets of partial label instances as well as assign valid labels to unlabeled instances. The label propagation procedure contains four phases. 1. Label propagation from partial label examples to unlabeled instances; 2. Label sets disambiguation of partial label examples; 3. Label set disambiguation of unlabeled instances; 4. Label propagation from unlabeled instances to partial label examples. The importance of unlabeled instances increases adaptively as the number of iteration increases, since they carry richer labeling information. Finally, unseen instances are classified based on the minimum reconstruction error on both partial label and unlabeled instances. Extensive experiments on real-world partial label

data sets clearly show that SSPL achieves highly competitive performance against state-of-the-art approaches.

The rest of this paper is organized as follows. Section 2 discusses existing works. Section 3 presents technical details of the proposed SSPL approach. Section 4 reports comparative experiments. Finally, Section 5 concludes.

2 Related Work

The problem focused in this paper, i.e. *semi-supervised partial label learning*, is the intersection of PLL and SLL. Both PLL and SSL can be regarded as *weakly supervised* frameworks [Zhou, 2017] where label information conveyed by training examples is implicit or partially inaccessible.

One kind of strategy attempts to fitting widely-used learning techniques to partial label examples. For maximum likelihood techniques, the likelihood function is defined as the probability of observing each partial label training example over its candidate label set [Liu and Dietterich, 2012]. For maximum margin techniques, the classification margin over each partial label training example is defined by discriminative modeling outputs from candidate labels and non-candidate labels [Nguyen and Caruana, 2008; Yu and Zhang, 2017]. For instance-based techniques, the candidate label sets of neighbouring instances are merged via weighted voting for making prediction [Hüllermeier and Beringer, 2006; Zhang and Yu, 2015]. Another kind of strategy aims to fit partial label examples to existing learning techniques. Following this, partial label examples can be transformed into binary examples via feature mapping [Chen *et al.*, 2014], one-vs-one decomposition [Wu and Zhang, 2018], or error-correcting outputs coded [Zhang *et al.*, 2017]. However, PLL methods are not sufficient to learn from semi-supervised partial label examples well, because they ignore the data distribution information lie in the large amount of unlabeled data which is known to be very useful.

SLL [Zhu and Goldberg, 2009; Li and Liang, 2019] aims to induce a classifier $f : \mathcal{X} \mapsto \mathcal{Y}$ from both labeled and unlabeled examples. There are four major categories of SLL approaches, i.e. generative methods [Miller and Uyar, 1997], graph-based methods [Blum and Chawla, 2001], low-density separation methods [Joachims, 1999] and disagreement-based methods [Zhou and Li, 2010]. Apparently, although SSL had taken the unlabeled instances into account, it still lacks of the ability of learning from partial label examples associated with candidate label sets.

Semi-supervised partial label learning is also related to other 'mixed' cases under weakly supervised learning frameworks such as *multi-instance multi-label learning* [Zhou *et al.*, 2012], *multi-instance active learning* [Settles *et al.*, 2008], *semi-supervised multi-label learning* [Kong *et al.*, 2013], *learning from incomplete and inaccurate supervision* [Zhang *et al.*, 2019] and *semi-supervised weak-label learning* [Dong *et al.*, 2018]. It is worth noting that the data scenario studied in this paper looks similar to the semi-supervised multi-label learning. However, multi-label learning focuses on exploiting the label relationship among providing labels while PLL aims to disambiguate them.

3 The Proposed Method

3.1 Problem Statement and Notation

In the original PLL, let $\mathcal{X} = \mathbb{R}^d$ be the d-dimensional instance space and $\mathcal{Y} = \{y_1, y_2, \ldots, y_q\}$ be the label space with q class labels. Formally, the partial label training set can be written as $\mathcal{D} = \{(\boldsymbol{x}_i, S_i) | 1 \leq i \leq m\}$, where $\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional feature vector $(x_{i1}, x_{i2}, \ldots, x_{id})^\mathsf{T}$ and $S_i \subseteq \mathcal{Y}$ is the associated candidate label set. Following the key assumption of PLL, the ground-truth label y_i for $\boldsymbol{x}_i$ is concealed in its candidate label set (i.e. $y_i \in S_i$) and therefore cannot be accessed by the learning algorithm.

In the *semi-supervised partial label learning*, the training set consists of not only partial label examples $\mathcal{D}_p = \{(\boldsymbol{x}_i, S_i) | 1 \leq i \leq m\}$ but also unlabeled instances $\mathcal{D}_u = \{\boldsymbol{x}_i | 1 \leq i \leq m\}$. Given the semi-supervised partial label training set $\mathcal{D} = \{\mathcal{D}_p \cup \mathcal{D}_u\}$, *semi-supervised partial label learning* aims to induce a classification model $f : \mathcal{X} \mapsto \mathcal{Y}$ from $\mathcal{D}$ such that for any unseen instance, f predicts its label.

3.2 A Simple Solution

The main difficulty of *semi-supervised partial label learning* lies in that the learning algorithm is required to disambiguating the candidate label sets of partial label examples and at the same time exploiting the data distribution information of unlabeled data. An intuitive solution is to disambiguate the candidate label sets of partial label training examples, i.e., picks up the valid single-label from candidate label set. Then, the problem is transformed into a simple SSL problem, which can be solved by a number of well-studied learning algorithms. Table 1 summarizes the procedure for such a simple solution.

Table 1: A simple solution of *semi-supervised partial label learning*.

Inputs:

$\mathcal{D}_p$: the partial label training set $\{(\boldsymbol{x}_i, S_i) \mid 1 \leq i \leq p\}$
$\mathcal{D}_u$: the unlabeled training set $\{\boldsymbol{x}_i \mid p+1 \leq i \leq p+u\}$
$\mathcal{P}$: PLL algorithm
$\mathcal{S}$: SSL learning algorithm
$\boldsymbol{x}^*$: the unseen instance

Outputs:

y^*: the predicted class label for $\boldsymbol{x}^*$

Process:

1: Disambiguate the partial label training examples $(\boldsymbol{x}_i, S_i)$ in $\mathcal{D}_p$ into single-label example $(\boldsymbol{x}_i, \hat{y}_i)$ with PLL algorithm $\hat{y} \leftarrow \mathcal{P}(\mathcal{D}_p)$;
2: Train a multi-class classifier with labeled and unlabeled examples by SSL algorithm $\mathcal{C} \leftarrow \mathcal{S}(\mathcal{D}_l, \mathcal{D}_u)$, here $\mathcal{D}_l$ denotes $\{(\boldsymbol{x}_i, \hat{y}_i) \mid 1 \leq i \leq p\}$;
3: Predict the unseen instance with $\mathcal{C}$.

In the algorithm above, the label set disambiguation procedure and the exploitation of unlabeled data are completely separated. The unlabeled instances can not help improving the disambiguation accuracy of partial label examples. To address this critical limitation, in the proposed SSPL approach, an iterative label propagation procedure between partial label

examples and unlabeled instances is employed to put them into a framework. In the following sections, the weighted graph construction procedure, which is necessary for label propagation, is first introduced and then the iterative label propagation algorithm.

3.3 Weighted Graph Construction

To realize the process of labeling information propagation from the source training set $\mathcal{S} = \{\boldsymbol{x}_i \mid 1 \leq i \leq s\}$ to the target training set $\mathcal{T} = \{\boldsymbol{x}_i \mid 1 \leq i \leq t\}$, a *weighted directed bipartite* graph $G = (U, V, E)$ is constructed over $\mathcal{S}$ and $\mathcal{T}$. While vertex set U corresponds to the instances in the source training set and vertex set V corresponds to the instances in target training set. Edge set E contains the directed edges from U to V (note that E only consists of this kind of edges but not, e.g., edges from U to U or from U to U). For each instance $\boldsymbol{x}_i$ in the target training set, its k-nearest neighbours $\mathcal{N}(\boldsymbol{x}_i)$ in the source training set are identified. Accordingly, the edges of graph G are set as $E = \{(\boldsymbol{x}_j, \boldsymbol{x}_i) \mid \boldsymbol{x}_j \in \mathcal{N}(\boldsymbol{x}_i), 1 \leq i \leq t\}$.

From the graph G constructed above, we can simply specify a $t \times s$ weight matrix $\mathbf{W} = [w_{i,j}]_{t \times s}$ where $w_{i,j} \geq 0$ if $(\boldsymbol{x}_j, \boldsymbol{x}_i) \in E$ and $w_{i,j} = 0$ otherwise. In order to capture the fine-grained influences between instances, in this paper, we employed the weights calculation method applied in the IPAL approach [Zhang and Yu, 2015], which determines the weights by solving an novel optimization problem (OP). Let $\hat{\boldsymbol{w}}_i = [w_{i,j_1}^{\hat{}}, \ldots, w_{i,j_k}^{\hat{}}]^{\mathsf{T}} \quad (j_a \in \mathcal{N}(x_i), 1 \leq a \leq k)$ denotes the weight vector of $\boldsymbol{x}_i$ in the target training set and its k-nearest neighbours $\mathcal{N}(x_i)$ in the source training set, the influences of the instances $\boldsymbol{x}_{j_a}$ in $\mathcal{N}(x_i)$ to $\boldsymbol{x}_i$ can be calculated by solving the following OP:

$$\begin{aligned} \min_{\boldsymbol{w}_i} \quad & \|\boldsymbol{x}_i - \sum_{a=1}^{k} w_{i,j_a} \cdot \boldsymbol{x}_{j_a}\| \\ s.t. \quad & w_{i,j_a} \geq 0 \quad (j_a \in \mathcal{N}(x_i), 1 \leq a \leq k) \end{aligned} \tag{1}$$

As shown in OP(1), the weight vector is optimized by fitting a linear least square problem subject to the non-negativity constraints, which can be obtained easily by a quadratic programming solver. Then the weight matrix $\mathbf{W}$ is normalized by row:$\tilde{\mathbf{W}} = \mathbf{D}^{-1}\mathbf{W}$. Here, $\mathbf{D} = diag[d_1, d_2, \ldots, d_t]$ is a diagonal matrix with $d_i = \sum_{j=1}^{s} w_{i,j}$. Table 2 summarizes the procedure of weighted graph construction.

3.4 Iterative Label Propagation

To facilitate the iterative label propagation procedure, four normalized weight matrix are constructed corresponding to the four phases in the label propagation procedure respectively. Specifically, $\mathbf{H} = \text{WGC}(\mathcal{D}_p, \mathcal{D}_u, k)$ is used for the label propagation from $\mathcal{D}_p$ (source training set) to $\mathcal{D}_u$ (target training set). $\mathbf{J} = \text{WGC}(\mathcal{D}_p, \mathcal{D}_p, k)$ is used for the label propagation from $\mathcal{D}_p$ to itself, i.e., label set disambiguation of partial label examples in $\mathcal{D}_p$. $\mathbf{K} = \text{WGC}(\mathcal{D}_u, \mathcal{D}_u, k)$ is used for the label propagation from $\mathcal{D}_u$ to itself. $\mathbf{L} = \text{WGC}(\mathcal{D}_u, \mathcal{D}_p, k)$ is used for the label propagation from $\mathcal{D}_u$ to $\mathcal{D}_p$.

In the SSPL approach, labeling confidence matrix for partial label training examples and unlabeled instances, i.e. $\mathbf{F_p} = [f_{i,c}]_{p \times q}$ and $\mathbf{F_u} = [f_{i,c}]_{u \times q}$ are introduced, where $f_{i,c} \geq 0$ corresponds to the probability of label y_c being the ground-truth label of instance $\boldsymbol{x}_i$. For partial label examples, the labeling confidence matrix can be initialized with $\mathbf{F_p} = \mathbf{P} = [p_{i,c}]_{p \times q}$.

$$1 \leq i \leq p: \quad p_{i,c} = \begin{cases} \dfrac{1}{|S_i|}, & \text{if } y_c \in S_i \\ 0\ , & \text{if } y_c \notin S_i \end{cases} \tag{2}$$

For unlabeled instances, the labeling confidence matrix can be initialized with $\mathbf{F_u} = \mathbf{U} = [u_{i,c}]_{u \times q}$.

$$1 \leq i \leq u: \quad u_{i,c} = \frac{1}{q} \tag{3}$$

In other words, at the initialization step, for partial label examples, the probability of a label being the ground-truth label of instance $\boldsymbol{x}_i$ is distributed over its candidate labels in S_i. And for unlabeled instances, the probability is distributed over all q labels in $\mathcal{Y}$.

The iterative label propagation procedure contains four phases: 1. label propagation from partial label examples to unlabeled instances; 2. label sets disambiguation of partial label examples; 3. label set disambiguation of unlabeled instances; 4. label propagation from unlabeled instances to partial label examples.

During the iteration, labeling confidence matrix $\mathbf{F_p}$ and $\mathbf{F_u}$ are updated according to the following equations.

$$\mathbf{F_u} = \alpha \cdot \mathbf{H} \cdot \mathbf{F_p} + (1 - \alpha) \cdot \mathbf{F_u} \tag{4}$$

$$\tilde{\mathbf{F}}_\mathbf{p} = \alpha \cdot \mathbf{J} \cdot \mathbf{F_p} + (1 - \alpha) \cdot \mathbf{P} \tag{5}$$

$$\mathbf{F_u} = \beta \cdot \mathbf{K} \cdot \mathbf{F_u} + (1 - \beta) \cdot \mathbf{F_u} \tag{6}$$

Table 2: WGC (weighted graph construction) procedure.

Inputs:

$\mathcal{S}$: the source training set $\{\boldsymbol{x}_i \mid 1 \leq i \leq s\}$

$\mathcal{T}$: the target training set $\{\boldsymbol{x}_i \mid 1 \leq i \leq t\}$

k: the number of nearest neighbour considered

Outputs:

$\tilde{\mathbf{W}}$: the normalized weight matrix

Process:

Initialize weight matrix $\mathbf{W} = [w_{i,j}]_{t \times s}$;
for $i = 1$ to t **do**
  Identify the k-nearest neighbours $\mathcal{N}(\boldsymbol{x}_i)$ in the source training set $\mathcal{S}$ for $\boldsymbol{x}_i$ in the target training set $\mathcal{T}$;
  Determine the weight vector $\hat{\boldsymbol{w}}_i = [w_{i,j_1}^{\hat{}}, \ldots, w_{i,j_k}^{\hat{}}]^{\mathsf{T}}$ w.r.t. $\boldsymbol{x}_j$ and $\mathcal{N}(\boldsymbol{x}_i)$ by solving OP(1);
  for $j_a \in \mathcal{N}(\boldsymbol{x}_i)$ **do**
    Set $w_{i,j_a} = w_{i,j_a}^{\hat{}}$;
  end for
end for
Normalize weight matrix $\mathbf{W}$ by column: $\tilde{\mathbf{W}} = \mathbf{D}^{-1}\mathbf{W}$.

Table 3: The proposed SSPL approach.

Inputs:

$\mathcal{D}_p$: the partial label training set $\{(\boldsymbol{x}_i, S_i) \mid 1 \leq i \leq p\}$

$\mathcal{D}_u$: the unlabeled training set $\{\boldsymbol{x}_i \mid p+1 \leq i \leq p+u\}$

k: the number of nearest neighbour considered

$\alpha, \tilde{\beta}, r$: the tuning parameter

T: the number of iterations

$\boldsymbol{x}^*$: the unseen instance

Outputs:

y^*: the predicted class label for $\boldsymbol{x}^*$

Process:

1: Construct normalized weight matrix $\mathbf{H}$, $\mathbf{J}$, $\mathbf{K}$ and $\mathbf{L}$ according to the WGC procedure in Table 2;
2: Initialize the label confidence matrix $\mathbf{F_p}$ and $\mathbf{F_u}$ according to Eq.(2) and Eq.(3);
3: **for** $t = 1$ to $\boldsymbol{T}$ **do**
4: Set $\mathbf{F_u}$ according to Eq.(4);
5: Set $\tilde{\mathbf{F}}_\mathbf{p}$ according to Eq.(5);
6: Re-scale $\tilde{\mathbf{F}}_\mathbf{p}$ to $\mathbf{F_p}$ according to Eq.(8);
7: Set $\mathbf{F_u}$ according to Eq.(6);
8: Set $\tilde{\mathbf{F}}_\mathbf{p}$ according to Eq.(7);
9: Re-scale $\tilde{\mathbf{F}}_\mathbf{p}$ to $\mathbf{F_p}$ according to Eq.(8);
10: **end for**
11: **for** $i = 1$ to p **do**
12: Disambiguate partial label example $(\boldsymbol{x}_i, S_i)$ into single label example $(\boldsymbol{x}_i, \hat{y_i})$ according to Eq.(9);
13: **end for**
14: **for** $i = p+1$ to $p+u$ **do**
15: Assign valid label $\hat{y_i}$ for unlabeled instance $\boldsymbol{x}_i$ according to Eq.(9);
16: **end for**
17: Identify the k-nearest neighbours of $\boldsymbol{x}^*$ in the $\mathcal{D}_p$ and $\mathcal{D}_u$, i.e. $\mathcal{N}_p(\boldsymbol{x}^*)$ and $\mathcal{N}_u(\boldsymbol{x}^*)$;
18: Determine the weight vectors $\boldsymbol{w}^*$ and $\boldsymbol{\theta}^*$ w.r.t. $\boldsymbol{x}^*$ and its k-nearest neighbours in $\mathcal{D}_p$ and $\mathcal{D}_u$ by solving OP(1);
19: Return the predicted class label y^* according to Eq.(10).

$$\tilde{\mathbf{F}}_\mathbf{p} = \beta \cdot \mathbf{L} \cdot \mathbf{F_u} + (1-\beta) \cdot \mathbf{F_p} \tag{7}$$

Here, parameters α and β are used to balance the importance of partial label examples and unlabeled instances. As partial label examples carry more labeling information than unlabeled ones i.e. more *important* than unlabeled instances, α is always larger than β. In the SSPL approach, parameter β is set to zero at the beginning of the iteration as the unlabeled instances carrying no labeling information. As the iteration goes on, the unlabeled instances carry more labeling information and β increases and finally approaches to the pre-defined upper bound $\tilde{\beta}$. Parameter α is set to be a constant.

The labeling confidence matrix of partial label examples, i.e., $\tilde{\mathbf{F}}_\mathbf{p}$, needs to be re-scaled into $\mathbf{F_p}$ after updated by propagating labeling information from $\tilde{\mathbf{F}}_\mathbf{p}$ and $\mathbf{F_u}$ after phase2 and phase4. The re-scale operation clears the probability of labels outside of the candidate label set being the ground-truth label of partial label examples and then normalize sum of the labeling probability of an instance into one.

$$1 \leq i \leq p : f_{i,c} = \begin{cases} \dfrac{\tilde{f_{i,c}}}{\sum_{y_l \in S_i} \tilde{f_{i,l}}}, & \text{if } y_c \in S_i \\ 0\ , & \text{if } y_c \notin S_i \end{cases} \tag{8}$$

Finally, we can pick up the valid labels of partial label and unlabeled instances based on the final labeling confidence matrix $\mathbf{F_p}$ and $\mathbf{F_u}$. Here, the SSPL approach adjusts the final label confidence towards the class prior distribution using the famous *class mass normalization* mechanism [Zhu and Goldberg, 2009]

$$\hat{y_i} = \arg\max_{y_c \in \mathcal{Y}} \ \frac{n_c}{\tilde{n_c}} \cdot f_{i,c} \tag{9}$$

Here, $n_c = \sum_{i=1}^{p} y_{i,c}$ is the prior distribution of y_c in the initial labeling confidence matrix P, and $\tilde{n_c} = \sum_{i=1}^{p+u} f_{i,c}$ is the class mass of y_c in the final labeling confidence matrix i.e. $\mathbf{F_p}$ and $\mathbf{F_u}$.

During the testing phase, the class label of an unseen instance $\boldsymbol{x}^*$ is predicted based on the minimum reconstruction error criterion on the disambiguated training examples, i.e. the disambiguated partial label examples $\{(\boldsymbol{x}_i, \hat{y_i}) \mid 1 \leq i \leq p\}$ and disambiguated unlabeled instances $\{(\boldsymbol{x}_i, \hat{y_i}) \mid p+1 \leq i \leq p+u\}$. Firstly, the k-nearest neighbours of $\boldsymbol{x}^*$ in $\mathcal{D}_p$ and $\mathcal{D}_u$ i.e. $\mathcal{N}_p(\boldsymbol{x}^*)$ and $\mathcal{N}_u(\boldsymbol{x}^*)$ are identified separately. After that, the weight vectors $\boldsymbol{w}^* = [w^*_{i_1}, w^*_{i_2}, \ldots, w^*_{i_k}]^\mathsf{T}$ $(i_a \in \mathcal{N}_p(\boldsymbol{x}^*), 1 \leq a \leq k)$ and $\boldsymbol{\theta}^* = [\theta^*_{i_1}, \theta^*_{i_2}, \ldots, \theta^*_{i_k}]^\mathsf{T}$ $(i_b \in \mathcal{N}_u(\boldsymbol{x}^*), 1 \leq a \leq k)$ are determined w.r.t. $\boldsymbol{x}^*$ and its k-nearest neighbours in $\mathcal{D}_p$ and $\mathcal{D}_u$ by solving the same optimization problem OP(1). Finally, the unseen instance is classified based on the following equation:

$$\begin{aligned} y^* = \arg\min_{y_c \in \mathcal{Y}} \ \| x^* - r \cdot \sum_{a=1}^{k} \mathbb{I}(\hat{y_{i_a}} = y_c) \cdot w^*_{i_a} \cdot \boldsymbol{x}^*_{i_a} \\ -(1-r) \cdot \sum_{b=1}^{k} \mathbb{I}(\hat{y_{i_b}} = y_c) \cdot \theta^*_{i_b} \cdot \boldsymbol{x}^*_{i_b} \| \end{aligned} \tag{10}$$

Here, r is the tuning parameter weighting the reconstruction error on partial label and unlabeled instances.

Table 3 summarizes the complete procedure of the proposed SSPL approach. Given the semi-supervised partial label training set, four normalized weight matrix are constructed over $\mathcal{D}_p$ and $\mathcal{D}_u$ (Step 1). After that, label confidence matrix $\mathbf{F_p}$ and $\mathbf{F_u}$ are initialized and updated by iterative label propagation (Steps 2-10). Then, the SSPL approach picks up the valid labels of training examples according to $\mathbf{F_p}$ and $\mathbf{F_u}$ (Steps 11-16). Finally, unseen instances are classified based on the minimum reconstruction error of disambiguated training examples (Steps 17-19).

4 Experiments

In this section, we compare the performances of the proposed SSPL approaches with several state-of-the-art PLL algorithms on various real-world tasks including: automatic face naming, object classification and bird song classification.

Table 4: Classification accuracy (mean±std) of each comparing algorithm on Lost, Soccer Player and Yahoo!News (from top to bottom). In addition, •/◦ indicates whether SSPL is statistically superior/inferior to the comparing algorithm on each data set (pairwise t-test at 5% significance level).

	$p = 0.05$	$p = 0.10$	$p = 0.15$	$p = 0.20$	$p = 0.30$	$p = 0.40$	$p = 0.50$
SSPL	0.299±0.004	0.374±0.003	0.408±0.002	0.461±0.002	0.489±0.003	0.529±0.003	0.556±0.001
IPAL	0.265±0.003	0.343±0.001	0.373±0.003	0.398±0.002	0.458±0.002	0.481±0.002	0.548±0.003
PL-KNN	0.193±0.001•	0.225±0.001•	0.254±0.001•	0.254±0.002•	0.267±0.001•	0.318±0.002•	0.335±0.001•
CLPL	0.308±0.001	0.384±0.001	0.380±0.002	0.363±0.001•	0.407±0.003	0.554±0.002	0.651±0.002◦
PL-SVM	0.127±0.003•	0.166±0.002•	0.262±0.003•	0.216±0.006•	0.253±0.002•	0.307±0.000•	0.302±0.005•

	$p = 0.05$	$p = 0.10$	$p = 0.15$	$p = 0.20$	$p = 0.30$	$p = 0.40$	$p = 0.50$
SSPL	0.493±0.000	0.500±0.000	0.500±0.000	0.506±0.000	0.510±0.000	0.517±0.000	0.523±0.000
IPAL	0.489±0.000	0.492±0.000	0.500±0.000	0.503±0.000	0.511±0.000	0.512±0.000	0.520±0.000
PL-KNN	0.489±0.000	0.491±0.000	0.491±0.000	0.490±0.000	0.491±0.000	0.492±0.000	0.492±0.000
CLPL	0.348±0.000•	0.280±0.001•	0.250±0.000•	0.254±0.000•	0.278±0.000•	0.331±0.000•	0.385±0.000•
PL-SVM	0.468±0.000•	0.465±0.001	0.439±0.007	0.465±0.001	0.478±0.000•	0.482±0.000•	0.480±0.000•

	$p = 0.05$	$p = 0.10$	$p = 0.15$	$p = 0.20$	$p = 0.30$	$p = 0.40$	$p = 0.50$
SSPL	0.428±0.000	0.474±0.000	0.497±0.000	0.514±0.000	0.536±0.000	0.556±0.000	0.566±0.000
IPAL	0.380±0.000•	0.437±0.000•	0.454±0.000•	0.477±0.000•	0.511±0.000•	0.529±0.000•	0.552±0.000•
PL-KNN	0.312±0.000•	0.351±0.000•	0.368±0.000•	0.375±0.000•	0.404±0.000•	0.424±0.000•	0.435±0.000•
CLPL	0.408±0.000•	0.465±0.000•	0.499±0.000	0.525±0.000	0.554±0.000◦	0.572±0.000	0.588±0.000◦
PL-SVM	0.245±0.004•	0.260±0.004•	0.230±0.004•	0.248±0.003•	0.243±0.003•	0.273±0.005•	0.251±0.001•

Table 5: Classification accuracy (mean±std) of each comparing algorithm on MSRCv2. In addition, •/◦ indicates whether SSPL is statistically superior/inferior to the comparing algorithm on each data set (pairwise t-test at 5% significance level).

	$p = 0.05$	$p = 0.10$	$p = 0.15$	$p = 0.20$	$p = 0.30$	$p = 0.40$	$p = 0.50$
SSPL	0.277±0.003	0.344±0.002	0.356±0.001	0.414±0.002	0.448±0.003	0.465±0.001	0.494±0.001
IPAL	0.289±0.003	0.337±0.002	0.349±0.003	0.386±0.005	0.398±0.000	0.444±0.001	0.446±0.001
PL-KNN	0.207±0.002	0.257±0.001•	0.276±0.002	0.317±0.001•	0.345±0.001•	0.357±0.001•	0.357±0.001•
CLPL	0.264±0.003	0.288±0.001•	0.287±0.002•	0.309±0.002•	0.326±0.001•	0.350±0.001•	0.361±0.001•
PL-SVM	0.169±0.002•	0.218±0.004•	0.240±0.003•	0.236±0.001•	0.222±0.001•	0.206±0.001•	0.204±0.001•

Table 7: Characteristic of the real-world partial label data sets.

Data Set	#Examples	#Features	#Class Labels	Avg. #CLs
Lost	1,122	108	16	2.23
MSRCv2	1,758	48	23	3.16
BirdSong	4,998	38	13	2.18
Soccer Player	17,472	279	171	2.09
Yahoo! News	22,991	163	219	1.91

4.1 Experimental Setup

The performance of SSPL is compared against five state-of-the-art PLL algorithms, each configured with parameters suggested in respective literature:

- IPAL [Zhang and Yu, 2015] disambiguate the candidate label set via an iterative label propagation procedure [suggested configuration: $k = 10, \alpha = 0.95, T = 100$]
- PL-KNN [Hüllermeier and Beringer, 2006] which adapts k-nearest neighbor technique to learn from partial label examples via weighted voting [suggested configuration: $k = 10$];
- CLPL [Cour *et al.*, 2011] which transforms PLL into binary learning problem via feature mapping with convex loss optimization [suggested configuration: SVM with squared hinge loss];
- PL-SVM [Nguyen and Caruana, 2008] which adapts maximum margin technique to learn from PL data via a l_2 regularization [suggested configuration: regularization parameter pool with $\{10^{-3}, \ldots, 10^{3}\}$];

For each data set, we consider the percentage of partial label examples in the whole training set by randomly sampling $p \in \{0.05, 0.10, 0.15, 0.20, 0.30, 0.40, 0.50\}$ instances from the whole training set with their candidate label sets and the other with no labeling information. For compared methods, only sampled partial label examples and their candidate label sets are provided. That is because PLL algorithms can not exploit from unlabeled data. According to the experiments conducted by us, using unlabeled data directly (simply view unlabeled instance as partial label example with its candidate label set equals to the whole label space $\mathcal{Y}$) always hurt the

Table 6: Classification accuracy (mean±std) of each comparing algorithm on BirdSong. In addition, •/◦ indicates whether SSPL is statistically superior/inferior to the comparing algorithm on each data set (pairwise t-test at 5% significance level).

	$p = 0.05$	$p = 0.10$	$p = 0.15$	$p = 0.20$	$p = 0.30$	$p = 0.40$	$p = 0.50$
SSPL	0.464±0.001	0.501±0.001	0.520±0.001	0.549±0.001	0.567±0.001	0.572±0.000	0.586±0.000
IPAL	0.443±0.001	0.474±0.000	0.497±0.002•	0.509±0.000•	0.538±0.001	0.554±0.000	0.557±0.000•
PL-KNN	0.333±0.001•	0.400±0.001•	0.426±0.000•	0.445±0.000•	0.469±0.000•	0.504±0.000•	0.518±0.000•
CLPL	0.347±0.001•	0.362±0.000•	0.359±0.000•	0.368±0.001•	0.365±0.000•	0.358±0.000•	0.357±0.000•
PL-SVM	0.230±0.005•	0.086±0.004•	0.151±0.009•	0.138±0.006•	0.178±0.012•	0.129±0.005•	0.182±0.012•

performance of PLL algorithms.

As shown in Table 3, parameters employed by SSPL are set as $k = 10$, $\alpha = 0.70$, $\tilde{\beta} = 0.25$, $r = 0.7$ and $T = 100$. In the rest of this section, five-fold cross-validation is performed on each real-world data set and in each training fold, the partial label instances are randomly sampled for three times. Accordingly, the mean predictive accuracies (and also the standard deviations) are recorded for all comparing algorithms.

4.2 Automatic Face Naming Task

In the automatic face naming task, our goal is to label the faces appeared in video or news images by the ground-truth names of characters or concerned people. For example, an episode of a video contains several characters and their faces may appear simultaneously in a screenshot. The scripts and dialogues are provided indicating which characters are in the screenshot, which forms the candidate label sets of the corresponding faces. Moreover, it is often the case that in a news collection every image is accompanied by a short textual description. Such a news image may contain several faces and the associated description will indicate the names of the people appeared in this image, which forms the candidate label sets of the faces. Three data sets are adopted for this task: Lost [Cour *et al.*, 2011], Soccer Player [Zeng *et al.*, 2013] and Yahoo!News [Guillaumin *et al.*, 2010], each contains 1122, 17472 and 22991 face images across 16, 171 and 219 classes respectively. The average amount of candidate label sets for a single label are 2.23, 2.09 and 1.91. Please refer to Table 7 for details.

Pairwise t-test at 0.05 significance level is conducted based on the five-fold cross-validation and three-time random sampling, where the test outcomes between SSPL and the comparing approaches are also recorded.

As shown in Table 4, we can observe that SSPL achieves highly comparable performances to all the state-of-the-art PLL algorithms. Furthermore, we can also find that: 1) On all the three data sets in this task, SSPL outperforms PL-KNN and PL-SVM at all subtasks; 2) SSPL achieves statistically superior performances to PL-KNN and PL-SVM on Lost and Yahoo!News and statistically superior to IPAL on Yahoo!News; 3) SSPL achieves comparable performances to CLPL on Lost and Yahoo!News while outperforms it on Soccer Player.

4.3 Object Classification Task

In this task, every image is segmented to several compact regions, and the labels of segmented regions, which are provided manually, forms the candidate label set of the image. Among the candidate label set, the label of the most dominant region is selected as the ground-truth single-label. Our goal is to predict the unseen images. MSRCv2 [Liu and Dietterich, 2012] data set is adopted for this task. This data set contains 1758 images with totally 23 classes. The average amount of candidate label sets for a single label is 3.16.

Results are presented in Table 5, from which we can observe that SSPL also achieves highly comparable performances to all the compared methods. Specifically, SSPL achieves the best performances on 6 of 7 subtasks (IPAL obtains the best performance when $p = 0.05$) and statistically superior to PL-KNN, CLPL and PL-SVM on most subtasks.

4.4 Bird Song Classification Task

The BirdSong [Briggs *et al.*, 2012] data set is adopted for this task. BirdSong is collected from 548 bird sound records. Each record is consisted of 1-40 syllables, leading to totally 4998 syllables included in the data set. The bird species appeared in every record are manually annotated, which forms the candidate label sets of every syllable. We need to identity which syllable is from which kind of bird.

Results are reported in Table 6. It is impressive to observe that the proposed SSPL approach significantly outperforms all the compared algorithms. The SSPL approach achieves the best performances on all subtasks. Moreover, SSPL is statistically superior to IPAL on 3 subtasks and superior to PL-KNN, CLPL and PL-SVM on all subtasks.

5 Conclusion

In this paper, we consider a new learning setup called *semi-supervised partial label learning* where the training set consists of two kinds of weak supervision, i.e., partial label data and unlabeled data. We propose an iterative label propagation method, which can process two kinds of weakly supervised data simultaneously by jointly propagating label between partial labeled and unlabeled instances, and derive a good label assignment. The experimental results show that our method is superior to approaches that only considering one kind of weak supervision. In future, we consider scaling up our model to large-scale data, and consider weak supervision data in dynamic environments.

Acknowledgments

This research was supported by the National Key R&D Program of China (2018YFB1004300) and the National Natural Science Foundation of China (61772262).

References

[Blum and Chawla, 2001] A. Blum and S. Chawla. Learning from labeled and unlabeled data using graph mincuts. In *Proceedings of the 18th International Conference on Machine Learning*, pages 19–26, Williamston, USA, 2001.

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

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

[Cour *et al.*, 2011] T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *Journal of Machine Learning Research*, 12:1501–1536, 2011.

[Dong *et al.*, 2018] H.-C. Dong, Y.-F. Li, and Z.-H. Zhou. Learning from semi-supervised weak-label data. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 2926–2933, New Orleans, USA, 2018.

[Guillaumin *et al.*, 2010] M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *Proceedings of the 11th European Conference on Computer Vision*, pages 634–647, Crete, Greece, 2010.

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

[Jie and Orabona, 2010] L. Jie and F. Orabona. Learning from candidate labeling sets. In *Advances in Neural Information Processing Systems 23*, pages 1504–1512. MIT Press, Cambridge, MA, 2010.

[Joachims, 1999] T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the 16th International Conference on Machine Learning*, pages 200–209, Bled, Slovenia, 1999.

[Kong *et al.*, 2013] X.-N. Kong, M. K. Ng, and Z.-H. Zhou. Transductive multi-label learning via label set propagation. *IEEE Transactions on Knowledge and Data Engineering*, 25(3):704–719, 2013.

[Li and Liang, 2019] Y.-F. Li and D.-M. Liang. Safe semi-supervised learning: A brief introduction. *Frontiers of Computer Science*, 2019.

[Liu and Dietterich, 2012] L. Liu and T. Dietterich. A conditional multinomial mixture model for superset label learning. In *Advances in Neural Information Processing Systems 25*, pages 557–565. MIT Press, Cambridge, MA, 2012.

[Miller and Uyar, 1997] D. J. Miller and H. S. Uyar. A mixture of experts classifier with learning based on both labelled and unlabelled data. In *Advances in neural information processing systems*, pages 571–577, 1997.

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

[Settles *et al.*, 2008] B. Settles, M. Craven, and S. Ray. Multiple-instance active learning. In *Advances in neural information processing systems 21*, pages 1289–1296, Vancouver, Canada, 2008.

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

[Wu and Zhang, 2018] X. Wu and M.-L. Zhang. Towards enabling binary decomposition for partial label learning. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 2868–2874, Stockholm, Sweden, 2018.

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

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

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

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

[Zhang *et al.*, 2019] Z.-Y. Zhang, P. Zhao, Y. Jiang, and Z.-H. Zhou. Learning from incomplete and inaccurate supervision. In *Proceedings of the 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Anchorage, AL, 2019.

[Zhou and Li, 2010] Z.-H. Zhou and Ming Li. Semi-supervised learning by disagreement. *Knowledge and Information Systems*, 24(3):415–439, 2010.

[Zhou *et al.*, 2012] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li. Multi-instance multi-label learning. *Artificial Intelligence*, 176(1):2291–2320, 2012.

[Zhou, 2017] Z.-H. Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2017.

[Zhu and Goldberg, 2009] X. Zhu and A. B. Goldberg. Introduction to semi-supervised learning. In *Synthesis Lectures to Artificial Intelligence and Machine Learning*, pages 1–130. Morgan & Claypool Publishers, San Francisco, CA, 2009.