Solving the Partial Label Learning Problem: An Instance-based Approach

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

Abstract

In partial label learning, each training example is associated with a set of *candidate* labels, among which only one is valid. An intuitive strategy to learn from partial label examples is to treat all candidate labels equally and make prediction by averaging their modeling outputs. Nonetheless, this strategy may suffer from the problem that the modeling output from the valid label is overwhelmed by those from the false positive labels. In this paper, an instance-based approach named IPAL is proposed by directly disambiguating the candidate label set. Briefly, IPAL tries to identify the valid label of each partial label example via an iterative label propagation procedure, and then classifies the unseen instance based on minimum error reconstruction from its nearest neighbors. Extensive experiments show that IPAL compares favorably against the existing instance-based as well as other state-of-the-art partial label learning approaches.

1 Introduction

Partial label learning deals with the problem where each training example is associated a set of candidate labels, among which only one label is assumed to be valid [Cour *et al.*, 2011; Zhang, 2014]. The problem of learning from partial label examples naturally arises in a number of real-world scenarios such as web mining [Jie and Orabona, 2010], multimedia contents analysis [Cour *et al.*, 2009; Zeng *et al.*, 2013], ecoinformatics [Liu and Dietterich, 2012], etc.[1]

Formally, let $\mathcal{X} = \mathbb{R}^d$ be the d-dimensional input space and $\mathcal{Y} = \{y_1, y_2, \ldots, y_q\}$ be the output space with q possible class labels. Given the partial label training set $\mathcal{D} = \{(\boldsymbol{x}_i, S_i) \mid 1 \leq i \leq m\}$, where $\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional feature vector $(x_{i1}, x_{i2}, \ldots, x_{id})^\top$ and $S_i \subseteq \mathcal{Y}$ is the set of candidate labels associated with $\boldsymbol{x}_i$, the task of partial label learning is to induce a *multi-class* classifier $f : \mathcal{X} \to \mathcal{Y}$ from $\mathcal{D}$. In partial label learning, the ground-truth label y_i for $\boldsymbol{x}_i$ is assumed to reside in the candidate label set (i.e. $y_i \in S_i$) but unknown to the learning algorithm.

As the ground-truth label is not accessible, one intuitive strategy to learn from partial label examples is to treat each candidate label in an equal manner for model induction [Cour *et al.*, 2011]. The final prediction is made by averaging the modeling outputs from all the candidate labels. However, one potential drawback of this strategy lies in that the essential output yielded by the ground-truth label (i.e. y_i) would be overwhelmed by the non-informative outputs yielded by the false positive labels (i.e. $S_i \setminus \{y_i\}$). Furthermore, the overwhelming effect caused by the false positive labels would be more pronounced as the size of candidate label set increases.

In this paper, rather than employing the above averaging strategy, we aim to solve the partial label learning problem by disambiguating the candidate label set directly. Accordingly, a novel partial label learning approach named IPAL, i.e. *Instance-based PArtial Label learning*, is proposed. Firstly, an asymmetric weighted graph over the training examples is constructed by affinity relationship analysis. After that, IPAL tries to identify the valid label of each partial label example via an iterative label propagation procedure. During the testing phase, the unseen instance is classified based on minimum error reconstruction from its nearest neighbors. Experimental studies on controlled UCI data sets as well as real-world partial label data sets clearly validate the effectiveness of IPAL against the comparing approaches.

The rest of this paper is organized as follows. Section 2 briefly discusses related work. Section 3 presents the technical details of the proposed IPAL approach. Section 4 reports the results of comparative experiments. Finally, Section 5 summarizes the paper and indicates future research issues.

2 Related Work

Partial label learning can be regarded as a *weakly-supervised* learning framework, where the supervision information conveyed by the partial label training examples are implicit. Conceptually speaking, it lies between two ends of the supervision spectrum, i.e. the traditional supervised learning with explicit supervision and the unsupervised learning with blind supervision. Partial label learning is related to other well-studied weakly-supervised learning frameworks, including *semi-supervised learning*, *multi-instance learning* and *multi-*

[1] In some cases, partial label learning is also termed as *ambiguous label learning* [Hüllermeier and Beringer, 2006; Chen *et al.*, 2014], *soft label learning* [Côme *et al.*, 2008] or *superset label learning* [Liu and Dietterich, 2014].

label learning. Nonetheless, different types of weak supervision information are handled by these learning frameworks.

Semi-supervised learning [Chapelle *et al.*, 2006; Zhu and Goldberg, 2009] learns from abundant unlabeled examples together with few labeled examples. For unlabeled data the ground-truth label assumes the whole label space, while for partial label data the ground-truth label is confined within the candidate label set. Multi-instance learning [Dietterich *et al.*, 1997; Amores, 2013] learns from labeled training examples each represented by a bag of instances. For multi-instance data the labels are assigned at the level of bags, while for partial label data the labels are assigned at the level of instances. Multi-label learning [Tsoumakas *et al.*, 2010; Zhang and Zhou, 2014] learns from training examples each associated with multiple labels. For multi-label data all the associated labels are valid ones, while for partial label data the associated labels are only candidate ones.

To learn from partial label examples, one intuitive strategy is to treat all the candidate labels in an equal manner and then average the outputs from all candidate labels for prediction. Following this strategy, a straightforward instance-based solution [Hüllermeier and Beringer, 2006] is to make prediction for unseen instance $\boldsymbol{x}^*$ in the following way: $f(\boldsymbol{x}^*) = \arg\max_{y\in\mathcal{Y}} \sum_{i\in\mathcal{N}(\boldsymbol{x}^*)} \mathbb{I}(y \in S_i)$. Here, $\mathbb{I}(\cdot)$ is the indicator function and the predicted label for $\boldsymbol{x}^*$ is determined by aggregating the votes from the candidate labels of its neighboring examples indexed in $\mathcal{N}(\boldsymbol{x}^*)$. Besides the instance-based instantiation, another solution following the averaging strategy is to assume a parametric model $F(\boldsymbol{x}, y; \boldsymbol{\theta})$ for discriminative learning [Cour *et al.*, 2011]. Here, the averaged output from all candidate labels, i.e. $\frac{1}{|S_i|}\sum_{y\in S_i} F(\boldsymbol{x}_i, y; \boldsymbol{\theta})$, is distinguished from the outputs from non-candidate labels, i.e. $F(\boldsymbol{x}_i, y; \boldsymbol{\theta})$ $(y \notin S_i)$.

Though the averaging 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 to learn from partial label examples is to disambiguate the candidate label set by identifying the ground-truth label. Existing approaches following this strategy view the ground-truth label as latent variable and make use of the Expectation-Maximization (EM) procedure [Dempster *et al.*, 1977] to refine the estimation of latent variable iteratively. The objective function optimized by the EM procedure can be instantiated based on the maximum likelihood criterion: $\sum_{i=1}^{m} \log\left(\sum_{y\in S_i} F(\boldsymbol{x}_i, y; \boldsymbol{\theta})\right)$ [Jin and Ghahramani, 2003; Grandvalet and Bengio, 2004; Liu and Dietterich, 2012], or the maximum margin criterion: $\sum_{i=1}^{m}\left(\max_{y\in S_i} F(\boldsymbol{x}_i, y; \boldsymbol{\theta}) - \max_{y\notin S_i} F(\boldsymbol{x}_i, y; \boldsymbol{\theta})\right)$ [Nguyen and Caruana, 2008].

In the next section, a novel partial label learning approach following the disambiguation strategy will be introduced. Different from EM-based disambiguation, the proposed approach does not assume any parametric model while tries to disambiguate the candidate label set by utilizing instance-based techniques.

3 The IPAL Approach

During the disambiguation phase, IPAL learns from partial label examples in two basic phases, i.e. *weighted graph construction* and *iterative label propagation*.

Let $\mathcal{D} = \{(\boldsymbol{x}_i, S_i) \mid 1 \leq i \leq m\}$ be the partial label training set, where $\boldsymbol{x}_i \in \mathcal{X}$ is a d-dimensional instance $(x_{i1}, x_{i2}, \ldots, x_{id})^\top$ and $S_i \subseteq \mathcal{Y}$ is the candidate label set associated with $\boldsymbol{x}_i$. In this paper, a weighted graph $G = (V, E)$ is constructed over the set of training examples with $V = \{\boldsymbol{x}_i \mid 1 \leq i \leq m\}$. For each instance $\boldsymbol{x}_i$, let $\mathcal{N}(\boldsymbol{x}_i)$ denote the indexes of its k-nearest neighbors identified in the training set, where the distance between two instances is calculated with the popular Euclidean metric. Accordingly, the edges of graph G are set as $E = \{(\boldsymbol{x}_i, \boldsymbol{x}_j) \mid i \in \mathcal{N}(\boldsymbol{x}_j),\ 1 \leq i \neq j \leq m\}$. In other words, there would be an (directed) edge from node $\boldsymbol{x}_i$ to node $\boldsymbol{x}_j$ iff $\boldsymbol{x}_i$ is among the k-nearest neighbors of $\boldsymbol{x}_j$.

From the set of edges E, one can simply specify an $m \times m$ weight matrix $\mathbf{W} = [w_{i,j}]_{m\times m}$ as follows: $w_{i,j} = 1$ if $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in E$; otherwise, $w_{i,j} = 0$. In order to encode fine-grained influences of neighboring instances, IPAL chooses to determine the weights by conducting some affinity relationship analysis. Let $\boldsymbol{w}_j = [w_{i_1,j}, w_{i_2,j}, \ldots, w_{i_k,j}]^\top$ $(i_a \in \mathcal{N}(\boldsymbol{x}_j), 1 \leq a \leq k)$ denote the weight vector w.r.t. $\boldsymbol{x}_j$ and its k-nearest neighbors, the influence of each neighboring instance $\boldsymbol{x}_{i_a}$ on $\boldsymbol{x}_j$ (i.e. $w_{i_a,j}$) is modeled by solving the following optimization problem (OP):

$$\begin{aligned} \min_{\boldsymbol{w}_j} \quad & \left\|\boldsymbol{x}_j - \sum\nolimits_{a=1}^{k} w_{i_a,j}\cdot \boldsymbol{x}_{i_a}\right\|^2 \\ \text{s.t.} \quad & w_{i_a,j} \geq 0 \quad (i_a \in \mathcal{N}(\boldsymbol{x}_j), 1 \leq a \leq k) \end{aligned} \tag{1}$$

As shown in OP (1), the weight vector $\boldsymbol{w}_j$ is optimized by fitting a linear least square problem subject to the non-negativity constraints. Here, we do not impose extra regularization term (e.g. L_1- or L_2-norm of $\boldsymbol{w}_j$) in the objective function to accommodate more space for optimization.

For OP (1), its optimal solution $\hat{\boldsymbol{w}}_j$ can be obtained by applying any off-the-shelf quadratic programming (QP) solver. To some extent, magnitude of the optimized weight $\hat{w}_{i_a,j}$ encodes the strength of affinity between $\boldsymbol{x}_j$ and its neighboring instance $\boldsymbol{x}_{i_a}$. Accordingly, IPAL specifies the weight matrix $\mathbf{W}$ as follows: $w_{i,j} = \hat{w}_{i,j}$ if $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in E$; otherwise $w_{i,j} = 0$. It is worth noting that $\mathbf{W}$ is an *asymmetric* weight matrix, which reflects the fact that the neighboring relationship is not necessarily symmetric. Furthermore, even when two instances $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ happen to be neighboring instance of each other, the influence from $\boldsymbol{x}_i$ to $\boldsymbol{x}_j$ (i.e. $w_{i,j}$) is generally different to that from $\boldsymbol{x}_j$ to $\boldsymbol{x}_i$ (i.e. $w_{j,i}$).[2]

To facilitate subsequent iterative label propagation procedure, the weight matrix $\mathbf{W}$ is then normalized by column: $\mathbf{H} = \mathbf{W}\mathbf{D}^{-1}$. Here, $\mathbf{D} = \mathrm{diag}[d_1, d_2, \ldots, d_m]$ is a diagonal matrix with $d_j = \sum_{i=1}^{m} w_{i,j}$. Let $\mathbf{F} = [f_{i,c}]_{m\times q}$ be an $m \times q$ matrix with non-negative entries, where $f_{i,c} \geq 0$ corresponds

[2] Therefore, symmetric setup of the weight matrix such as $w_{i,j} = \exp\left(-\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||^2}{2\sigma^2}\right)$ [Zhou *et al.*, 2004; Zhu and Goldberg, 2009] won't suffice under this circumstance.

to the labeling confidence of y_c being the ground-truth label of $\boldsymbol{x}_i$. Based on the partial label training set, an initial (prior) labeling confidence matrix $\mathbf{F}^{(0)} = \mathbf{P} = [p_{i,c}]_{m \times q}$ can be instantiated as follows:

$$\forall\, 1 \le i \le m: \quad p_{i,c} = \begin{cases} \frac{1}{|S_i|}, & \text{if } y_c \in S_i \\ 0\ , & \text{otherwise} \end{cases} \tag{2}$$

In other words, at the initialization step, the labeling confidence of $\boldsymbol{x}_i$ is equally distributed over its candidate labels in S_i. At the t-th iteration, $\mathbf{F}$ is updated by propagating labeling information along with the normalized weight matrix $\mathbf{H}$:

$$\tilde{\mathbf{F}}^{(t)} = \alpha \cdot \mathbf{H}^\top \mathbf{F}^{(t-1)} + (1-\alpha) \cdot \mathbf{P} \tag{3}$$

Here, parameter $\alpha \in (0,1)$ controls the relative amount of information inherited from label propagation and initial labeling. After that, $\tilde{\mathbf{F}}^{(t)}$ is re-scaled into $\mathbf{F}^{(t)}$ by consulting the candidate label set of each training example:

$$\forall\, 1 \le i \le m: \quad f_{i,c}^{(t)} = \begin{cases} \frac{\tilde{f}_{i,c}^{(t)}}{\sum_{y_l \in S_i} \tilde{f}_{i,l}^{(t)}}, & \text{if } y_c \in S_i \\ 0\ \ , & \text{otherwise} \end{cases} \tag{4}$$

As the iterative procedure terminates, one can disambiguate each partial label training example $(\boldsymbol{x}_i, S_i)$ based on the final labeling confidence matrix $\hat{\mathbf{F}}$ as follows: $\hat{y}_i = \arg\max_{y_c \in \mathcal{Y}} \hat{f}_{i,c}$. In this paper, IPAL further adopts the *class mass normalization* (CMN) mechanism [Zhu and Goldberg, 2009] to adjust the disambiguation output towards class prior distribution:

$$\hat{y}_i = \arg\max_{y_c \in \mathcal{Y}} \frac{n_c}{\hat{n}_c} \cdot \hat{f}_{i,c} \tag{5}$$

Here, $n_c = \sum_{i=1}^m p_{i,c}$ is the class mass of y_c w.r.t. prior labeling confidence matrix $\mathbf{P}$, and $\hat{n}_c = \sum_{i=1}^m \hat{f}_{i,c}$ is the class mass of y_c w.r.t. final labeling confidence matrix $\hat{\mathbf{F}}$.

During the testing phase, the class label of an unseen instance $\boldsymbol{x}^*$ is predicted based on the disambiguated training examples $(\boldsymbol{x}_i, \hat{y}_i)$. The k-nearest neighbors of $\boldsymbol{x}^*$ in the training set, i.e. $\mathcal{N}(\boldsymbol{x}^*)$, are firstly identified. After that, the weight vector $\boldsymbol{w}^* = [w^*_{i_1}, w^*_{i_2}, \ldots, w^*_{i_k}]^\top$ $(i_a \in \mathcal{N}(\boldsymbol{x}^*), 1 \le a \le k)$ w.r.t. $\boldsymbol{x}^*$ and its k-nearest neighbors are determined by solving the same optimization problem as shown in OP (1) (replacing $\{\boldsymbol{x}_j, \boldsymbol{w}_j\}$ with $\{\boldsymbol{x}^*, \boldsymbol{w}^*\}$). Thereafter, the unseen instance is classified based on the following minimum error reconstruction criterion:

$$y^* = \arg\min_{y_c \in \mathcal{Y}} \left\| \boldsymbol{x}^* - \sum\nolimits_{a=1}^{k} \mathbb{I}(\hat{y}_{i_a} = y_c) \cdot w^*_{i_a} \cdot \boldsymbol{x}_{i_a} \right\| \tag{6}$$

Table 1 summarizes the complete procedure of the proposed IPAL approach. Given the partial label training set, an asymmetric weighted graph are constructed by conducting affinity relationship analysis between each instance and its k-nearest neighbors (Steps 1-8). After that, an iterative label propagation procedure is performed to disambiguate the candidate label set of each training example (Steps 9-19). Finally, the unseen instance is classified based on minimum error reconstruction from its k-nearest neighbors (Steps 20-22).

Table 1: The pseudo-code of IPAL.

```
Inputs:
  $\mathcal{D}$ : the partial label training set $\{(\boldsymbol{x}_i, S_i) \mid 1 \le i \le m\}$
       $(\boldsymbol{x}_i \in \mathcal{X}, S_i \subseteq \mathcal{Y}, \mathcal{X} = \mathbb{R}^d, \mathcal{Y} = \{y_1, y_2, \ldots, y_q\})$
  $k$ : the number of nearest neighbors considered
  $\alpha$ : the balancing coefficient in $(0,1)$
  $T$ : the number of iterations
  $\boldsymbol{x}^*$ : the unseen instance
Outputs:
  $y^*$ : the predicted class label for $\boldsymbol{x}^*$
Process:
1:  Initialize weight matrix $\mathbf{W} = [w_{i,j}]_{m \times m}$ with $w_{i,j} = 0$;
2:  for $j = 1$ to $m$ do
3:      Identify the $k$-nearest neighbors $\mathcal{N}(\boldsymbol{x}_j)$ in $\mathcal{D}$ for $\boldsymbol{x}_j$;
4:      Determine the weight vector $\hat{\boldsymbol{w}}_j = [\hat{w}_{i_1,j}, \ldots, \hat{w}_{i_k,j}]^\top$
        w.r.t. $\boldsymbol{x}_j$ and its $k$-nearest neighbors by solving OP (1);
5:      for $i_a \in \mathcal{N}(\boldsymbol{x}_j)$ do
6:          Set $w_{i_a,j} = \hat{w}_{i_a,j}$;
7:      end for
8:  end for
9:  Normalize weight matrix $\mathbf{W}$ by column: $\mathbf{H} = \mathbf{W}\mathbf{D}^{-1}$;
10: Set the initial labeling confidence matrix $\mathbf{P}$ according to
    Eq.(2);
11: Set $\mathbf{F}^{(0)} = \mathbf{P}$;
12: for $t = 1$ to $T$ do
13:     Set $\tilde{\mathbf{F}}^{(t)}$ according to Eq.(3);
14:     Re-scale $\tilde{\mathbf{F}}^{(t)}$ into $\mathbf{F}^{(t)}$ according to Eq.(4);
15: end for
16: Set the final labeling confidence matrix $\hat{\mathbf{F}} = \mathbf{F}^{(T)}$;
17: for $i = 1$ to $m$ do
18:     Disambiguate partial label example $(\boldsymbol{x}_i, S_i)$ into
        single-label example $(\boldsymbol{x}_i, \hat{y}_i)$ according to Eq.(5);
19: end for
20: Identify the $k$-nearest neighbors $\mathcal{N}(\boldsymbol{x}^*)$ in $\mathcal{D}$ for $\boldsymbol{x}^*$;
21: Determine the weight vector $\boldsymbol{w}^* = [w^*_{i_1}, \ldots, w^*_{i_k}]^\top$ w.r.t.
    $\boldsymbol{x}^*$ and its $k$-nearest neighbors by solving OP (1);
22: Return the predicted class label $y^*$ according to Eq.(6).
```

4 Experiments

4.1 Experimental Setup

In this paper, two series of comparative experiments are conducted on controlled UCI data sets [Bache and Lichman, 2013] as well as real-world partial label data sets. Table 2 summarizes characteristics of these experimental data sets.

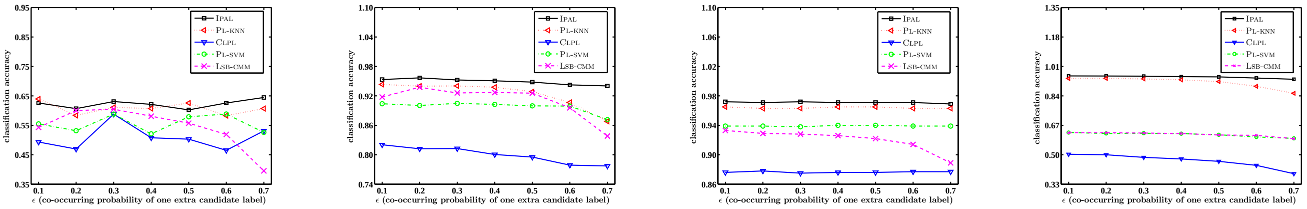
Following the popular controlling protocol [Cour *et al.*, 2011; Liu and Dietterich, 2012; Zhang, 2014; Chen *et al.*, 2014], an artificial partial label data set is generated from a multi-class UCI data set under different configurations of three controlling parameters p, r and ϵ. Here, p controls the proportion of examples which are partially labeled (i.e. $|S_i| > 1$), r controls the number of false positive labels in the candidate label set (i.e. $|S_i| = r + 1$), and ϵ controls the co-occurring probability between one extra candidate label and the ground-truth label. A total of 28 (4x7) parameter configurations considered in this paper are listed in Table 2.

The real-world partial label data sets are collected from

Table 2: Characteristics of the experimental data sets.

Controlled UCI Data Sets				`Configurations`
Data set	# Examples	# Features	# Class Labels	
glass	214	10	5	(I) $p = 1, r = 1, \epsilon \in \{0.1, 0.2, \ldots, 0.7\}$ **[Figure 1]**
segment	2,310	18	7	(II) $r = 1, p \in \{0.1, 0.2, \ldots, 0.7\}$ **[Figure 2]**
usps	9,298	256	10	(III) $r = 2, p \in \{0.1, 0.2, \ldots, 0.7\}$ **[Figure 3]**
letter	20,000	16	26	(IV) $r = 3, p \in \{0.1, 0.2, \ldots, 0.7\}$ **[Figure 4]**

Real-World Data Sets					
Data set	# Examples	# Features	# Class Labels	Avg. # CLs	Domain
Lost	1122	108	16	2.23	*automatic face naming* [Cour *et al.*, 2011]
MSRCv2	1,758	48	23	3.16	*object classification* [Liu and Dietterich, 2012]
BirdSong	4,998	38	13	2.18	*bird song classification* [Briggs *et al.*, 2012]
Soccer Player	17,472	279	171	2.09	*automatic face naming* [Zeng *et al.*, 2013]
Yahoo! News	22,991	163	219	1.91	*automatic face naming* [Guillaumin *et al.*, 2010]



(a) `glass` (b) `segment` (c) `usps` (d) `letter`

Figure 1: Classification accuracy of each comparing algorithm changes as ϵ (co-occurring probability of one extra candidate label) increases (with 100% partially labeled examples $[p = 1]$ and one false positive candidate label $[r = 1]$).

several application domains including `Lost` [Cour *et al.*, 2011], `Soccer Player` [Zeng *et al.*, 2013] and `Yahoo! News` [Guillaumin *et al.*, 2010] for automatic face naming from images or videos, `MSRCv2` [Liu and Dietterich, 2012] for object classification, and `BirdSong` [Briggs *et al.*, 2012] for bird song classification. The average number of candidate labels (Avg. #CLs) for each real-world data set is also listed in Table 2.

The performance of IPAL is compared against four state-of-the-art partial label learning algorithms, each configured with parameters suggested in respective literature:

- PL-KNN [Hüllermeier and Beringer, 2006]: an instance-based approach to partial label learning by adopting the averaging strategy, where the number of nearest neighbors considered by PL-KNN is set to be 10;
- CLPL [Cour *et al.*, 2011]: a parametric approach to partial label learning by adopting the averaging strategy, where the parametric model is set to be SVM with squared hinge loss;
- PL-SVM [Nguyen and Caruana, 2008]: a maximum margin approach to partial label learning by adopting the EM-based disambiguation strategy, where the regularization parameter is chosen among $\{10^{-3}, \ldots, 10^{3}\}$ via cross-validation;
- LSB-CMM [Liu and Dietterich, 2012]: a maximum likelihood approach to partial label learning by adopting the EM-based disambiguation strategy, where the number of mixture components is set to be the number of class labels of each data set.

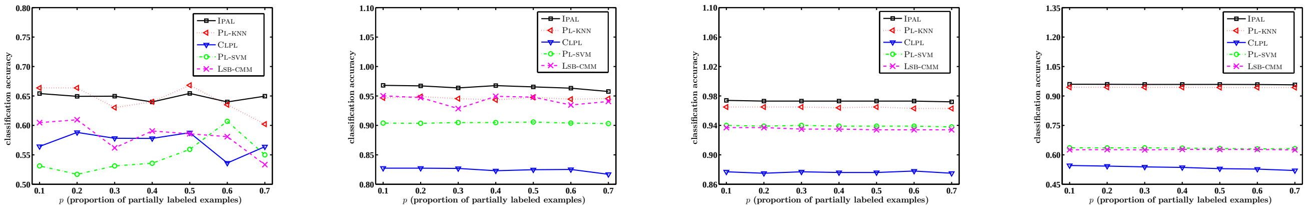
As shown in Table 1, parameters employed by IPAL are set as $k = 10$, $\alpha = 0.95$ and $T = 100$.[3] In the rest of this section, ten-fold cross-validation is performed on each artificial as well as real-world partial label data set. Accordingly, the mean predictive accuracies (and also the standard deviations) are recorded for all comparing algorithms.

4.2 Experimental Results

Controlled UCI Data Sets

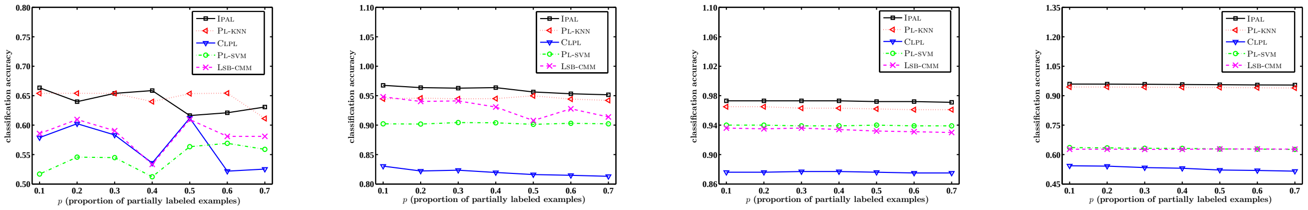
Figure 1 illustrates the classification accuracy of each comparing algorithm as the co-occurring probability ϵ varies from 0.1 to 0.7 with step-size 0.1 ($p = 1, r = 1$). One label y' is designated as the extra candidate label for each class label $y \in \mathcal{Y}$, where y' is chosen to co-occur with y with probability ϵ when y is the ground-truth label. Otherwise, any other class label would be chosen to co-occur with y. Figures 2 to 4 illustrate the classification accuracy of each comparing algorithm as the proportion p varies from 0.1 to 0.7 with step-size 0.1 ($r = 1, 2, 3$). For any partially labeled example, along with the ground-truth label, r class labels in $\mathcal{Y}$ will be randomly picked up to constitute the candidate label set.

[3] Sensitivity analysis on IPAL's parameter configuration is reported in Subsection 4.3.



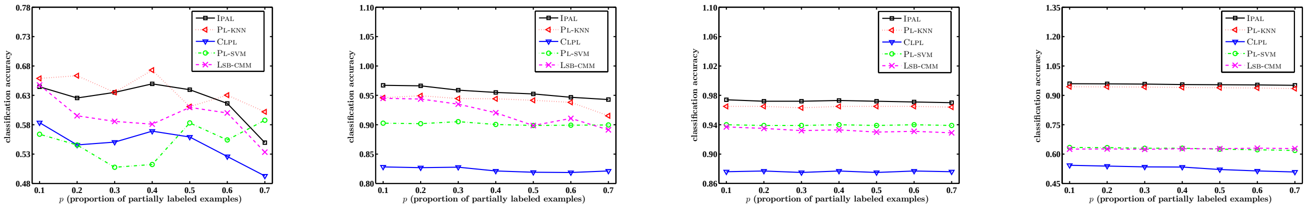
(a) glass (b) segment (c) usps (d) letter

Figure 2: Classification accuracy of each comparing algorithm changes as p (proportion of partially labeled examples) increases (with one false positive candidate label $[r = 1]$).



(a) glass (b) segment (c) usps (d) letter

Figure 3: Classification accuracy of each comparing algorithm changes as p (proportion of partially labeled examples) increases (with two false positive candidate labels $[r = 2]$).



(a) glass (b) segment (c) usps (d) letter

Figure 4: Classification accuracy of each comparing algorithm changes as p (proportion of partially labeled examples) increases (with three false positive candidate labels $[r = 3]$).

Table 3: Win/tie/loss counts (pairwise t-test at 0.05 significance level) on the classification performance of IPAL against other comparing algorithms.

	IPAL against			
	PL-KNN	CLPL	PL-SVM	LSB-CMM
[Figure 1]	**22/4/2**	**24/4/0**	**28/0/0**	**23/5/0**
[Figure 2]	**24/0/4**	**28/0/0**	**28/0/0**	**23/5/0**
[Figure 3]	**23/1/4**	**28/0/0**	**28/0/0**	**22/6/0**
[Figure 4]	**22/1/5**	**26/2/0**	**27/0/1**	**22/6/0**
In Total	**91/6/15**	**106/6/0**	**111/0/1**	**90/22/0**

As illustrated in Figures 1 to 4, the performance of IPAL is highly competitive to other comparing algorithms in most cases. Specifically, pairwise t-test at 0.05 significance level is conducted based on the results of ten-fold cross-validation. Table 3 summarizes the win/tie/loss counts between IPAL and the comparing algorithms. Out of the 112 statistical comparisons (28 configurations × 4 UCI data sets), it is shown that: 1) IPAL achieves superior performance against PL-KNN in 81.2% cases and has been outperformed by PL-KNN in only 13.4% cases; 2) IPAL achieves superior performance against CLPL and LSB-CMM in 94.6% and 80.3% cases respectively, and is comparable to both of them in the rest cases; 3) IPAL is shown to be inferior to PL-SVM in only 1 out of 112 cases, and outperforms PL-SVM in the rest cases.

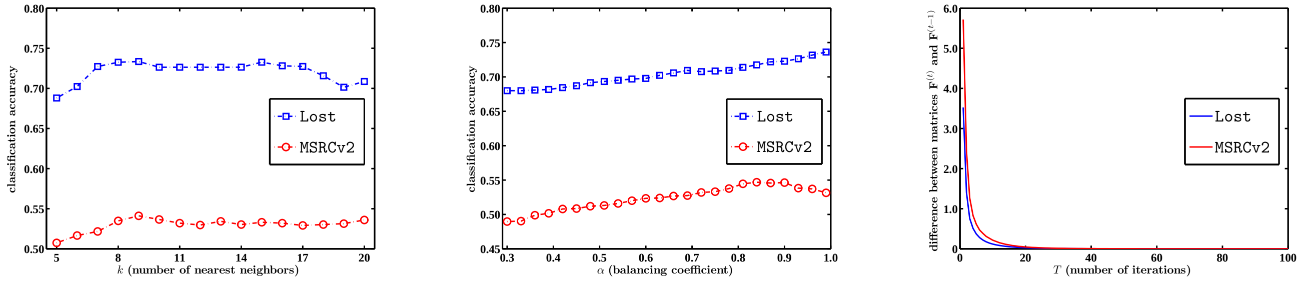
Real-World Data Sets

Table 4 reports the performance of each comparing algorithm on the real-world partial label data sets. Pairwise t-test at 0.05 significance level is conducted based on the results of ten-fold cross-validation, and the test outcomes between IPAL and other comparing algorithms are recorded.

As shown in Table 4, it is impressive that no algorithm has outperformed IPAL on the real-world data sets. Furthermore, we can also observe that: 1) IPAL achieves superior performance against PL-KNN on all data sets, which is also an instance-based approach to learning from partial label exam-

Table 4: Classification accuracy (mean± std. deviation) of each comparing algorithm on the real-world partial label data sets. In addition, •/○ indicates whether IPAL is statistically superior/inferior to the comparing algorithm on each data set (pairwise t-test at 0.05 significance level).

	IPAL	PL-KNN	CLPL	PL-SVM	LSB-CMM
Lost	0.726±0.041	0.388±0.036•	0.742±0.038	0.729±0.040	0.707±0.055
MSRCv2	0.523±0.025	0.445±0.030•	0.413±0.039•	0.482±0.043•	0.456±0.031•
BirdSong	0.708±0.014	0.649±0.021•	0.632±0.017•	0.663±0.032•	0.717±0.024
Soccer Player	0.547±0.014	0.492±0.015•	0.368±0.010•	0.443±0.014•	0.525±0.015•
Yahoo! News	0.667±0.006	0.456±0.010•	0.462±0.009•	0.636±0.010•	0.648±0.007



(a) Varying k $(\alpha = 0.95, T = 100)$ (b) Varying α $(k = 10, T = 100)$ (c) Varying T $(k = 10, \alpha = 0.95)$

Figure 5: Parameter sensitivity analysis for IPAL on the `Lost` and `MSRCv2` data sets. (a) Classification accuracy of IPAL changes as the number of nearest neighbors k increases from 5 to 20 with step-size 1; (b) Classification accuracy of IPAL changes as the balancing coefficient α increases from 0.30 to 0.99 with step-size 0.03; (c) Difference between two adjacent labeling confidence matrix (measured by L_2 norm $||\mathbf{F}^{(t)} - \mathbf{F}^{(t-1)}||$) converges with increasing number of iterations.

ples following the averaging strategy; 2) IPAL is comparable to CLPL and PL-SVM on the `Lost` data set, and achieves superior performance than both of them on the rest data sets; 3) IPAL is comparable to LSB-CMM on the `Lost`, `BirdSong` and `Yahoo! News` data sets, and achieves superior performance than LSB-CMM on the other two data sets.

4.3 Sensitivity Analysis

According to Table 1, IPAL learns from partial label examples by employing three parameters, i.e. k (number of nearest neighbors), α (balancing coefficient) and T (number of iterations). To study the sensitivity of IPAL w.r.t. them, Figure 5 illustrates how IPAL performs under different parameter configurations. For clarity of illustration, `Lost` and `MSRCv2` are employed here for analysis purpose while similar observations can be made on other data sets. As shown in Figure 5, it is obvious that:

- The performance of IPAL improves slightly as k increases from 5 and becomes stable shortly after k reaches 8 (Figure 5(a));
- The performance of IPAL improves steadily as α increases from 0.3 (Figure 5(b)). These observations indicate that the amount of labeling information propagated from neighboring instances (i.e. the first term of Eq.(3)) plays a key role for the effectiveness of IPAL;
- The model of IPAL (labeling confidence matrix $\mathbf{F}$) changes significantly in initial label propagation iterations and becomes convergent when T reaches 20 (Figure 5(c)).

Therefore, the parameter configuration specified for IPAL in Subsection 4.1 ($k = 10$, $\alpha = 0.95$, $T = 100$) naturally follows from the above analysis.

5 Conclusion

In this paper, the problem of partial label learning is studied where an instance-based approach named IPAL is proposed. Instead of employing the averaging strategy, IPAL aims to learn from partial label examples by directly disambiguating the candidate label via iterative label propagation. Extensive comparative studies clearly validate the effectiveness of IPAL.

In terms of label propagation, an important future work is to explore other ways to construct the weighted graph. For instance-based approach, it is also interesting to investigate better distance metric other than Euclidean distance for k-nearest neighbors identification.

References

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

[Bache and Lichman, 2013] K. Bache and M. Lichman. UCI machine learning repository. School of Information and Computer Sciences, University of California, Irvine, 2013.

[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.

[Chapelle *et al.*, 2006] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

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

[Côme *et al.*, 2008] E. Côme, L. Oukhellou, T. Denoeux, and P. Aknin. Mixture model estimation with soft labels. In D. Dubois, M. A. Lubiano, H. Prade, M. A. Gil, P. Grzegorzewski, and O. Hryniewicz, editors, *Advances in Soft Computing 48*, pages 165–174. Springer, Berlin, 2008.

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

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

[Dempster *et al.*, 1977] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B*, 39(1):1–38, 1977.

[Dietterich *et al.*, 1997] T. G. Dietterich, R. H. Lathrop, and T. Lozano-Pérez. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 89(1-2):31–71, 1997.

[Grandvalet and Bengio, 2004] Y. Grandvalet and Y. Bengio. Learning from partial labels with minimum entropy. Technical report, Center for Interuniversity Research and Analysis of Organizations, Québec, Canada, 2004.

[Guillaumin *et al.*, 2010] M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. In K. Daniilidis, P. Maragos, and N. Paragios, editors, *Lecture Notes in Computer Science 6311*, pages 634–647. Springer, Berlin, 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 J. Lafferty, C. K. I. Williams, J. Shawe-Taylor, R. S. Zemel, and A. Culotta, editors, *Advances in Neural Information Processing Systems 23*, pages 1504–1512. MIT Press, Cambridge, MA, 2010.

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

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

[Liu and Dietterich, 2014] L. Liu and T. Dietterich. Learnability of the superset label learning problem. In *Proceedings of the 31st International Conference on Machine Learning*, pages 1629–1637, Beijing, China, 2014.

[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.

[Tsoumakas *et al.*, 2010] G. Tsoumakas, I. Katakis, and I. Vlahavas. Mining multi-label data. In O. Maimon and L. Rokach, editors, *Data Mining and Knowledge Discovery Handbook*, pages 667–686. Springer, Berlin, 2010.

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

[Zhang, 2014] M.-L. Zhang. Disambiguation-free partial label learning. In *Proceedings of the 14th SIAM International Conference on Data Mining*, pages 37–45, Philadelphia, PA, 2014.

[Zhou *et al.*, 2004] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf. Learning with local and global consistency. In S. Thrun, L. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 284–291. MIT Press, Cambridge, MA, 2004.

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