Towards Mitigating the Class-Imbalance Problem for Partial Label Learning

Jing Wang
[†]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[‡]Key Lab. of Computer Network and Information Integration (Southeast University), MOE, China
jing_w@seu.edu.cn

Min-Ling Zhang*
[†]School of Computer Science and Engineering, Southeast University, Nanjing 210096, China
[‡]Key Lab. of Computer Network and Information Integration (Southeast University), MOE, China
[♯]Collaborative Innovation Center for Wireless Communications Technology, China
zhangml@seu.edu.cn

ABSTRACT

Partial label (PL) 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 valid. It is well-known that the problem of class-imbalance stands as a major factor affecting the generalization performance of multi-class classifier, and this problem becomes more pronounced as the ground-truth label of each PL training example is not directly accessible to the learning approach. To mitigate the negative influence of class-imbalance to partial label learning, a novel class-imbalance aware approach named CIMAP is proposed by adapting over-sampling techniques for handling PL training examples. Firstly, for each PL training example, CIMAP disambiguates its candidate label set by estimating the confidence of each class label being ground-truth one via weighted k-nearest neighbor aggregation. After that, the original PL training set is replenished for model induction by over-sampling existing PL training examples via manipulation of the disambiguation results. Extensive experiments on artificial as well as real-world PL data sets show that CIMAP serves as an effective data-level approach to mitigate the class-imbalance problem for partial label learning.

CCS CONCEPTS

• **Computing methodologies** → **Supervised learning**; **Machine learning algorithms**;

KEYWORDS

Partial label learning, Class-imbalance, Over-sampling

*Corresponding author

KDD '18, August 19–23, 2018, London, United Kingdom

ACM ISBN 978-1-4503-5552-0/18/08…$15.00
https://doi.org/10.1145/3219819.3220008

ACM Reference Format:
Jing Wang and Min-Ling Zhang. 2018. Towards Mitigating the Class-Imbalance Problem for Partial Label Learning. In *KDD '18: The 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, August 19–23, 2018, London, United Kingdom.* ACM, New York, NY, USA, Article 4, 10 pages. https://doi.org/10.1145/3219819.3220008

1 INTRODUCTION

In partial label (PL) learning, each training example is represented by a single instance while associated with a set of *candidate* labels, among which only one class label is valid [9, 31]. Formally, let $\mathcal{X} = \mathbb{R}^d$ be the d-dimensional instance space and $\mathcal{Y} = \{\lambda_1, \lambda_2, \ldots, \lambda_q\}$ be the label space with q class labels. Given the PL training set $\mathcal{D} = \{(\boldsymbol{x}_i, S_i) \mid 1 \leq i \leq m\}$, partial label learning aims to induce a *multi-class* classification model $f : \mathcal{X} \mapsto \mathcal{Y}$ from $\mathcal{D}$. For each PL training example $(\boldsymbol{x}_i, S_i)$, $\boldsymbol{x}_i \in \mathcal{X}$ corresponds to a d-dimensional feature vector $(x_{i1}, x_{i2}, \ldots, x_{id})^\top$ and $S_i \subseteq \mathcal{Y}$ corresponds to the candidate label set associated with $\boldsymbol{x}_i$. The key assumption of partial label learning lies in that the ground-truth label y_i for $\boldsymbol{x}_i$ is concealed in its candidate label set, i.e. $y_i \in S_i$.

In recent years, partial label learning techniques have shown to be successful to solve real-world tasks involving weakly-supervised information, such as web mining [17], image classification [6, 8, 28], ecoinformatics [20, 31], etc. Specifically, most existing partial label learning approaches learn from PL training examples by trying to maximize the classification accuracy of the predictive model [6, 7, 9, 11, 20, 21, 27, 29]. Considering that the goal of partial label learning is to induce a multi-class classifier, the problem of *class-imbalance* which widely exists for multi-class classification will have significant influence on the performance of the learning approach. As shown in Table 3, the imbalance ratio between the two classes with most and least number of examples ranges from 10.63 to 48.03 across the real-world PL data sets. Under the class-imbalance scenario, accuracy maximization does not serve as a good choice for model induction as the accuracy metric tends to overlook minority classes with less examples.

Class-imbalance learning for multi-class classification has been well investigated, where algorithm-level approaches work by amending specific learning techniques to fit imbalanced

data distribution [15, 19, 24, 25] while data-level approaches work by manipulating the training set to facilitate subsequent model induction [1, 10, 23]. However, the class-imbalance problem in partial label learning is more challenging as the ground-truth label for each PL training example is not accessible to the learning approach. Existing approaches to multi-class imbalanced problem rely on the explicit labeling information of training examples to enable the learning procedure, e.g. by under-/over-sampling training examples having specific class labels, which makes them not applicable to learn from PL examples.

In this paper, a first attempt towards addressing the class-imbalance problem for partial label learning is presented. A novel approach named CIMAP, i.e. *Class-IMbalance Aware Partial label learning*, is proposed accordingly by customizing the over-sampling strategy. For each PL training example, its candidate label set is firstly disambiguated by estimating the confidence of each class label being ground-truth one via k-nearest neighbor aggregation. After that, CIMAP replenishes the original PL training set for subsequent model induction where three over-sampling methods are applied by manipulating the disambiguation results. To show the effectiveness of the proposed data-level class-imbalance learning approach, experimental studies on both artificial and real-world PL data sets are conducted over four well-established partial label learning algorithms. Comprehensive evaluation results clearly validate that, in terms of class-imbalance aware metrics, the performance of partial label learning algorithm can be significantly improved by incorporating the class-imbalance mitigation scheme of CIMAP.

The rest of this paper is organized as follows. Section 2 briefly reviews related works on partial label learning. Section 3 presents technical details of the proposed CIMAP approach. Section 4 reports experimental results across different data sets, PL learning algorithms and class-imbalance aware metrics. Finally, Section 5 concludes.

2 RELATED WORK

Partial label learning can be regarded as a *weakly-supervised* learning framework [32] where the labeling information conveyed by training examples are implicit to the learning algorithm. It is related to other popular weakly-supervised learning frameworks such as *semi-supervised learning*, *multi-instance learning* and *multi-label learning*. However, the weak supervision nature of these learning frameworks is attributed to different forms of training examples to be dealt with, i.e. PL example with implicit supervision for partial label learning [9], unlabeled example with blind supervision for semi-supervised learning [34], multi-instance example with ambiguous supervision for multi-instance learning [2], and multi-label example with non-unique supervision for multi-label learning [33].

One solution to partial label learning is to enable canonical learning techniques with the ability of handling PL training examples. For maximum likelihood techniques, the likelihood function is defined as the probability of observing each PL training example over its candidate label set [18, 20]. For k-nearest neighbor techniques, the candidate label sets of neighboring instances are synergized to make final prediction on unseen instance [11, 16, 29]. For maximum margin techniques, the classification margin is defined over the predictive difference between candidate labels and non-candidate labels of each PL training example [21, 27]. Another solution to partial label learning is to transform PL examples into other forms so as to accommodate traditional learning settings. To accommodate binary learning setting, PL training examples can be transformed into binary examples via feature mapping [9], one-vs-one decomposition [26], or error-correcting output codes [30]. To accommodate multi-class learning setting, PL training examples can be transformed into multi-class examples via dictionary matching [7].

The problem of class-imbalance has been well investigated in multi-class classification, where a number of approaches have been proposed from algorithm-level or data-level perspectives. Algorithm-level approaches work by amending the training procedure of specific learning techniques to take class-imbalance characteristics into consideration, such as utilizing skew-insensitive Hellinger distance splitting criterion for decision tree building [15], introducing cost matrix or enhancing ensemble diversity to instantiate weight updates for AdaBoost [24, 25], choosing training examples dynamically for each updating epoch of neural networks [19]. Data-level approaches work by manipulating the training set to make subsequent multi-class model induction procedure feasible, mostly via over-sampling the training examples from minority classes [1, 10, 23].

Existing approaches to class-imbalance learning assume the availability of explicit labeling information from training examples, which is not the case under the partial label learning scenario. In the next section, the first data-level approach towards class-imbalance aware partial label learning is proposed.

3 THE PROPOSED APPROACH

The CIMAP approach consists of two phases to accomplish data-level manipulation of the PL training set, including *candidate label set disambiguation* and *training set replenishment*.

In the first phase, CIMAP aims to disambiguate the candidate label set of each PL training example serving as the basis for follow-up replenishment phase. Based on the notations given in Section 1, for each PL example $(\boldsymbol{x}, S)$, let $\boldsymbol{b}^S = [b_1^S, \ldots, b_q^S]^\top$ denote the q-dimensional binary vector w.r.t. the candidate label set:

$$\forall\ 1 \le j \le q:\ b_j^S = \begin{cases} 1, & \text{if } \lambda_j \in S \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

Accordingly, let $\mathcal{N}(\boldsymbol{x})$ denote the index set of k nearest neighbors identified for $\boldsymbol{x}$ among the training examples in $\mathcal{D}$. Then,

a real-valued vector $\boldsymbol{\gamma}_i = [\gamma_{i1}, \ldots, \gamma_{iq}]^\top$ is generated to characterize the confidence of each class label being the ground-truth one for the i-th training instance $\boldsymbol{x}_i$:

$$\boldsymbol{\gamma}_i = \sum\nolimits_{j \in \mathcal{N}(\boldsymbol{x}_i)} \left(1 - \frac{d(\boldsymbol{x}_i, \boldsymbol{x}_j)}{\sum_{k \in \mathcal{N}(\boldsymbol{x}_i)} d(\boldsymbol{x}_i, \boldsymbol{x}_k)}\right) \cdot \boldsymbol{b}^{S_j} \quad (2)$$

Here, $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ corresponds to the distance between $\boldsymbol{x}_i$ and one of its k nearest neighbors $\boldsymbol{x}_j$. Therefore, the confidence vector can be regarded as a weighted voting over the binary vectors of the neighboring examples. Specifically, the voting weight is inversely proportional to $d(\boldsymbol{x}_i, \boldsymbol{x}_j)$ which is measured by Euclidean distance in this paper.

After obtaining the confidence vector for each PL training example, CIMAP performs candidate label set disambiguation based on the $m \times q$ confidence matrix $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2, \ldots, \boldsymbol{\gamma}_m]^\top$. Firstly, a multi-class data set $\mathcal{M}$ is derived by disambiguating along each row of $\boldsymbol{\Gamma}$:

$$\mathcal{M} = \bigcup\nolimits_{j=1}^{q} G_j \quad (3)$$
$$\text{where } G_j = \{(\boldsymbol{x}_i, \lambda_j) \mid 1 \le i \le m,\ j = \arg\max_{1 \le k \le q} \gamma_{ik}\}$$

Conceptually, G_j stores the training examples whose disambiguated class label corresponds to λ_j. To ensure the feasibility of follow-up replenishment phase, CIMAP makes further adjustment on $\mathcal{M}$ by enforcing a threshold constraint on the size of G_j. For any data set G_j whose size is smaller the threshold parameter, i.e. $|G_j| < \tau$, a total of $\tau - |G_j|$ examples will be progressively added to G_j by traversing the j-th column of $\boldsymbol{\Gamma}$ in descending order of γ_{ij}. Here, the traversing procedure is conducted by transferring examples from other data sets G_k ($k \neq j$) into G_j without compromising their own threshold constraints.

In the second phase, CIMAP aims to replenish the original PL training set making them amenable for subsequent model induction. Let $j^* = \arg\max_{1 \le j \le q} |G_j|$ denote the index of the disambiguated data set with largest number of training examples. By referring to the generated multi-class data set $\mathcal{M}$, CIMAP employs three implementations of over-sampling techniques to fulfill the replenishment task:

- *Random over-sampling* (ROS): For each class label $\lambda_j \neq \lambda_{j^*}$, one example $(\boldsymbol{x}_i, \lambda_j) \in G_j$ is randomly sampled from the j-th disambiguated data set and the original PL training set is replenished with a new PL example as: $\mathcal{D} = \mathcal{D} \bigcup \{(\boldsymbol{x}_i, S_i)\}$. Note that for the sampled instance $\boldsymbol{x}_i$, its candidate label set S_i instead of its disambiguated class label λ_j is used for PL training set replenishment.
- *Synthetic over-sampling* (SMOTE): Following the synthetic over-sampling procedure of SMOTE [5, 14], one example $(\boldsymbol{x}_i, \lambda_j)$ is randomly sampled from G_j and let $(\boldsymbol{x}_r, \lambda_j)$ be the example randomly selected from $\boldsymbol{x}_i$'s k nearest neighbors in G_j. Then, a new synthetic instance $\hat{\boldsymbol{x}}_i = [\hat{x}_{i1}, \hat{x}_{i2}, \ldots, \hat{x}_{id}]^\top$ is generated by interpolating $\boldsymbol{x}_i$ and $\boldsymbol{x}_r$:
$$\hat{x}_{ia} = x_{ia} + (x_{ra} - x_{ia}) \cdot \omega_a \quad (1 \le a \le d) \quad (4)$$
Here, $\boldsymbol{\omega} = [\omega_1, \omega_2, \ldots, \omega_d]^\top$ is a random vector with each component ω_a taking value in [0, 1]. Accordingly, the original PL training set is replenished with a new PL example as: $\mathcal{D} = \mathcal{D} \bigcup \{(\hat{\boldsymbol{x}}_i, S_i)\}$.
- *Perturbation over-sampling* (POS): Similar to the above SMOTE process, let $(\boldsymbol{x}_i, \lambda_j)$ be the example randomly sampled from G_j and $(\boldsymbol{x}_r, \lambda_j)$ be one of its randomly selected k nearest neighbor in G_j. Furthermore, let $\boldsymbol{\omega}$ be a random vector with component values in [0, 1]. In addition to interpolate $\boldsymbol{x}_i$ and $\boldsymbol{x}_r$ within instance space, their associated candidate label set S_i and S_r are also interpolated via the following perturbation operation:
$$\hat{S}_i = \{\lambda_j \mid \hat{b}_j = 1,\ 1 \le j \le q\}, \text{ where} \quad (5)$$
$$\hat{\boldsymbol{b}} = \text{sign}\left[\boldsymbol{b}^{S_i} + \left((\boldsymbol{x}_r - \boldsymbol{x}_i)^\top \boldsymbol{\omega}\right) \cdot (\boldsymbol{b}^{S_r} - \boldsymbol{b}^{S_i})\right]$$
Here, the sign($\boldsymbol{z}$) function returns a binary vector by thresholding each component of $\boldsymbol{z}$ at 0. Accordingly, the original PL training set is replenished with a new PL example as: $\mathcal{D} = \mathcal{D} \bigcup \{(\hat{\boldsymbol{x}}_i, \hat{S}_i)\}$.

For each of the above over-sampling implementations, the replenishment procedure is repeated for $|G_{j^*}| - |G_j|$ times w.r.t. each class label λ_j other than λ_{j^*}. The CIMAP approach instantiated with each of the over-sampling implementation is denoted as CIMAP-ROS, CIMAP-SMOTE, and CIMAP-POS respectively.

Table 1 summarizes the complete procedure of CIMAP. Given the PL training set, the confidence vector for each PL training example is firstly estimated based on k-nearest neighbor aggregation (Steps 1-4). After that, a multi-class disambiguation data set is generated to reflect class-imbalance characteristics (Steps 5-14). As guided by the generated multi-class data set, the original PL training set is replenished via specific over-sampling strategy (Steps 15-26). Finally, the replenished PL data set is used to induce the multi-class classification model which makes prediction on the unseen instance (Steps 27-28).

Generally, there are several properties which are noteworthy for the proposed CIMAP approach:

Firstly, CIMAP serves as a *data-level* approach to addressing the class-imbalance issue for partial label learning. Although it is feasible to directly apply some multi-class imbalance learning algorithm upon the disambiguated multi-class data set $\mathcal{M}$ (after Step 14) to induce the predictive model $f : \mathcal{X} \mapsto \mathcal{Y}$, CIMAP turns to leverage $\mathcal{M}$ as an intermediate source of information to manipulate the PL training set. In this way, the class-imbalance issue is addressed by keeping the partial label nature of training examples and thus any off-the-shelf PL learning algorithm can be coupled with the improved PL training set for model induction (Step 27).

Secondly, as a first attempt towards class-imbalance aware partial label learning, the goal of CIMAP is to manifest one possible solution to the investigated problem while those technical choices adopted by CIMAP are by no means meant to be optimal. For instance, the k-nearest neighbor strategy [16] has been employed to help estimate the confidence

Table 1: The pseudo-code of CIMAP.

Inputs:
$\mathcal{D}$: the partial label training set $\{(\boldsymbol{x}_i, S_i) \mid 1 \le i \le m\}$
k: the number of nearest neighbors considered
τ: the threshold parameter
ν: the over-sampling mode $\nu \in$ {ROS, SMOTE, POS}
$\mathcal{L}$: the PL learning algorithm for model induction
$\boldsymbol{x}$: the unseen instance

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

Process:
1: **for** $i = 1$ to m **do**
2: Identify the index set of k nearest neighbors $\mathcal{N}(\boldsymbol{x}_i)$ for $\boldsymbol{x}_i$ in PL training set $\mathcal{D}$;
3: Generate the confidence vector $\boldsymbol{\gamma}_i$ w.r.t. Eq.(2);
4: **end for**
5: Derive the multi-class disambiguation data set $\mathcal{M} = \{G_1, \ldots, G_q\}$ according to Eq.(3);
6: **for** $j = 1$ to q **do**
7: $\mathcal{I} = \{i \mid 1 \le i \le m,\ (\boldsymbol{x}_i, \lambda_j) \notin G_j\}$;
8: **while** $|G_j| < \tau$ **do**
9: Identify $i' = \arg\max_{i \in \mathcal{I}} \gamma_{ij}$ and the corresponding class label $\lambda_{j'} \in \mathcal{Y}$ with $(\boldsymbol{x}_{i'}, \lambda_{j'}) \in G_{j'}$;
10: **if** $|G_{j'}| > \tau + 1$ **then**
11: $G_j = G_j \bigcup \{(\boldsymbol{x}_{i'}, \lambda_j)\}$, $G_{j'} = G_{j'} \setminus \{(\boldsymbol{x}_{i'}, \lambda_{j'})\}$, $\mathcal{I} = \mathcal{I} \setminus \{i'\}$;
12: **end if**
13: **end while**
14: **end for**
15: Identify $j^* = \arg\max_{1 \le j \le q} |G_j|$;
16: **for** $j \in \{1, 2, \ldots, q\} \setminus j^*$ **do**
17: $count = 0$;
18: **while** $count < |G_{j^*}| - |G_j|$ **do**
19: **switch** ν **do**
20: **case** ROS: Randomly sample $(\boldsymbol{x}_i, \lambda_j) \in G_j$ and replenish $\mathcal{D} = \mathcal{D} \bigcup \{(\boldsymbol{x}_i, S_i)\}$;
21: **case** SMOTE: Randomly sample $(\boldsymbol{x}_i, \lambda_j) \in G_j$, generate synthetic instance $\hat{\boldsymbol{x}}_i$ w.r.t. Eq.(4) and replenish $\mathcal{D} = \mathcal{D} \bigcup \{(\hat{\boldsymbol{x}}_i, S_i)\}$;
22: **case** POS: Randomly sample $(\boldsymbol{x}_i, \lambda_j) \in G_j$, generate perturbed label set $\hat{S}_i$ w.r.t Eq.(5) and replenish $\mathcal{D} = \mathcal{D} \bigcup \{(\hat{\boldsymbol{x}}_i, \hat{S}_i)\}$;
23: **endswitch**
24: $count = count + 1$;
25: **end while**
26: **end for**
27: Induce multi-class classifier f by invoking $\mathcal{L}$ on the replenished PL data set $\mathcal{D}$, i.e. $f \leftarrow\!\shortmid \mathcal{L}(\mathcal{D})$;
28: Return $y = f(\boldsymbol{x})$.

vector (Eq.(2)) which can otherwise be fulfilled with alternative instance-based estimation strategies [11, 29, 31], the over-sampling strategy has been employed to help replenish the PL training set which can otherwise be fulfilled with under-sampling or adaptive sampling [14].

Thirdly, for the multi-class disambiguation data set $\mathcal{M} = \{G_1, \ldots, G_q\}$, each disambiguated example $(\boldsymbol{x}_i, \lambda_j) \in G_j$ is not enforced to satisfy the partial label assumption, i.e. $\lambda_j \in S_i$. It is not difficult to satisfy this assumption by replacing the condition $j = \arg\max_{1 \le k \le q} \gamma_{ik}$ with $j = \arg\max_{\lambda_k \in S_i} \gamma_{ik}$ in Eq.(3) and adding an extra condition $\lambda_j \in S_{i'}$ in Step 10 of Table 1. Nonetheless, CIMAP chooses the simplified strategy of constraint-free disambiguation so as to lower the risk of overfitting PL training examples.

4 EXPERIMENTAL RESULTS

4.1 Experimental Setup

4.1.1 Data Sets. To thoroughly evaluate the effectiveness of the proposed CIMAP approach, two series of experiments are conducted with one on controlled UCI data sets [3] and the other on real-world PL data sets. Tables 2 and 3 summarize the characteristics of artificial and real-world PL data sets respectively, where the imbalance ratio between the largest and smallest classes (IR) as well as the distribution over all classes are also included.[1]

Following the widely-used controlling protocol [7, 9, 20, 30], an artificial PL data set can be derived from one multi-class UCI data set by configuring 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. As shown in Table 2, a total of 28 (4x7) parameter configurations have been considered for each controlled UCI data set.

The real-world PL data sets are collected from several application domains including FG-NET [22] for facial age estimation, Lost [9], Soccer Player [28] and Yahoo!News [12] for automatic face naming from images or videos, MSRCv2 [20] for object classification, and BirdSong [4] for bird song classification. For *facial age estimation*, human faces are represented as instances while ages annotated by crowdsourcing labelers are regarded as candidate labels. For *automatic face naming*, faces cropped from an image or video frame are represented as instances while names extracted from the associated captions or subtitles are regarded as candidate labels. For *object classification*, image segmentations are represented as instances while objects appearing within the same image are regarded as candidate labels. For *bird song classification*, singing syllables of the birds are represented as instances while bird species jointly singing are regarded as candidate labels. The average number of candidate labels (avg. #CLs) for each real-world PL data set is also recorded in Table 3.

[1] As a common practice in class-imbalance studies [14], the case of *extreme imbalance* is not considered in this paper. Specifically, any class label leading to overly-high imbalance ratio (IR>50) is excluded from the label space.

Table 2: Characteristics of the controlled UCI data sets.

Data Set	#Examples	#Features	#Class Labels	IR	Class Distribution
Glass	214	9	6	8.44	*76/70/29/17/13/9*
Ecoli	332	7	6	28.60	*143/77/52/35/20/5*
Abalone	4,153	7	19	49.21	*689/634/568/487/391/267/259/203/126/115/103/67/58/57/42/32/26/15/14*

`Configurations`

(I) $r = 1$, $p \in \{0.1, 0.2, \ldots, 0.7\}$ (II) $r = 2$, $p \in \{0.1, 0.2, \ldots, 0.7\}$

(III) $r = 3$, $p \in \{0.1, 0.2, \ldots, 0.7\}$ (IV) $p = 1$, $r = 1$, $\epsilon \in \{0.1, 0.2, \ldots, 0.7\}$

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

Data Set	#Examples	#Features	#Class Labels	avg. #CLs	IR	Cls. Dist.	Task Domain
FG-NET	1,002	262	63	7.34	47.00	`(a)`	*facial age estimation* [22]
Lost	1,122	108	14	2.22	11.33	`(b)`	*automatic face naming* [9]
MSRCv2	1,755	48	22	3.15	10.63	`(c)`	*object classification* [20]
BirdSong	4,998	38	13	2.18	40.00	`(d)`	*bird song classification* [4]
Soccer Player	8,883	279	170	1.77	25.44	`(e)`	*automatic face naming* [28]
Yahoo! News	17,262	163	17	1.85	48.03	`(f)`	*automatic face naming* [12]

`(a)` *47/43/42/42/41/41/40/40/39/38/37/37/33/32/31/30/28/27/25/23/22/20/19/17/17/16/12/11/11/11/9/9/9/9/8/8/7/6/6/6/5/5/4/4/4/3/3/3/3/3/3/2/2/2/2/2/1/1/1/1/1/1*

`(b)` *204/198/142/103/103/88/76/61/33/26/25/25/20/18*

`(c)` *255/187/182/175/160/87/77/76/63/61/51/48/46/39/37/34/32/32/31/31/27/24*

`(d)` *1280/810/602/501/494/345/277/190/139/126/120/82/32*

`(e)` *229/166/151/148/148/146/142/140/126/123/111/105/96/93/91/88/85/82/81/79/79/78/77/76/74/74/74/73/72/70/68/68/67/67/66/65/65/65/65/65/64/63/63/63/63/62/62/62/61/61/61/61/60/60/59/59/58/58/57/57/57/56/56/56/56/55/55/54/54/53/53/53/52/52/52/51/50/50/49/49/48/48/48/48/47/47/46/46/46/45/45/45/42/42/42/42/42/42/41/41/41/40/40/40/40/40/39/38/38/38/37/36/36/36/36/35/34/33/33/32/32/31/31/31/31/31/30/30/29/29/29/28/27/27/26/25/25/25/24/24/24/23/23/22/22/22/21/20/19/19/19/18/18/18/17/16/16/16/15/15/15/15/14/14/13/12/12/11/11/9*

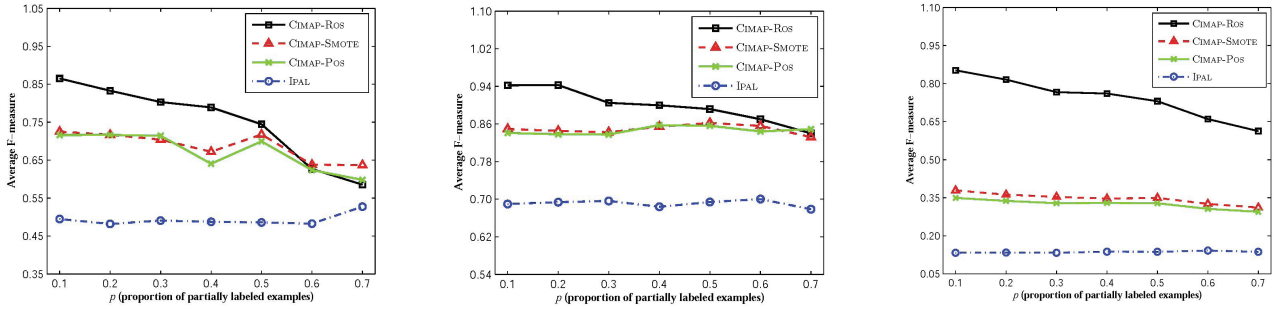
`(f)` *4323/4227/3167/2150/1168/469/312/262/231/167/124/112/109/109/98/94/90*

Table 4: Win/tie/loss counts (pairwise t-test at 0.05 significance level) on the coupling PL learning algorithm and its CIMAP variants in terms of each class-imbalance aware metric.

Evaluation metric	`CIMAP-ROS` vs. coupling algorithm				`CIMAP-SMOTE` vs. coupling algorithm				`CIMAP-POS` vs. coupling algorithm			
	PL-KNN	PL-SVM	CLPL	IPAL	PL-KNN	PL-SVM	CLPL	IPAL	PL-KNN	PL-SVM	CLPL	IPAL
Avg. Precision	60/23/1	79/5/0	65/19/0	72/12/0	62/22/0	49/35/0	55/28/1	77/6/1	44/40/0	48/36/0	41/43/0	72/12/0
Avg. Recall	53/29/2	78/6/0	56/28/10	68/15/1	60/23/1	47/37/0	43/41/0	77/7/0	41/42/1	41/43/0	45/39/0	70/14/0
Avg. F-measure	54/25/5	79/5/0	65/19/0	62/20/2	60/22/2	54/30/0	52/30/2	72/12/0	48/33/3	54/30/0	52/30/2	67/17/0
MAUC	40/43/1	60/24/0	52/31/1	6/50/28	41/40/3	49/35/0	44/40/0	70/14/0	40/41/3	55/29/0	42/42/0	69/15/0

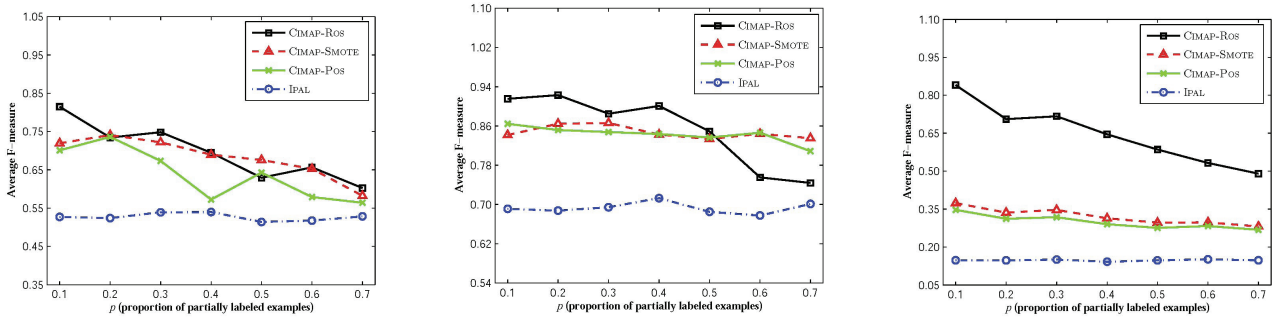
4.1.2 Performance Metrics. For multi-class classifier, accuracy serves as the commonly-used metric for performance evaluation. Nonetheless, under class-imbalance scenario, it is inappropriate to employ this metric as accuracy is insensitive to how the classifier works on minority classes with less examples. In this paper, several class-imbalance aware metrics are employed for performance evaluation which balance how the classifier works on majority as well as minority classes [1, 15, 25].

Given the multi-class test set $\mathcal{T} = \{(\boldsymbol{x}_i, y_i) \mid 1 \le i \le n\}$ with n_j $(1 \le j \le q)$ examples belonging to the j-th class, i.e. $n = \sum_{j=1}^{q} n_j$. Furthermore, let $\mathbf{P} = [p_{ij}]_{n \times q}$ be the $n \times q$ output matrix where p_{ij} corresponds to predictive confidence of $\boldsymbol{x}_i$ having the j-th class label λ_j such that $f(\boldsymbol{x}_i) = \arg\max_{1 \le j \le q} p_{ij}$. Accordingly, let $\mathbf{C} = [c_{jk}]_{q \times q}$ be the confusion matrix where c_{jk} stores the number of examples from λ_j which are actually classified as λ_k based on f. Thereafter, the performance of f on λ_j can be characterized by the following quantities: a) `Precision`: $P_j = \frac{c_{jj}}{\sum_{k=1}^{q} c_{kj}}$; b) `Recall`: $R_j = \frac{c_{jj}}{n_j}$; c) `F-measure`: $F_j = \frac{2P_j \cdot R_j}{P_j + R_j}$; d) `Pairwise AUC`: the AUC A_{jk} between λ_j and λ_k calculated from the



(a) Glass (b) Ecoli (c) Abalone

Figure 1: Average F-measure of the coupling algorithm IPAL and its CIMAP variants changes as p (proportion of partially labeled examples) increases from 0.1 to 0.7 (with one false positive candidate label $[r = 1]$).



(a) Glass (b) Ecoli (c) Abalone

Figure 2: Average F-measure of the coupling algorithm IPAL and its CIMAP variants changes as p (proportion of partially labeled examples) increases from 0.1 to 0.7 (with two false positive candidate label $[r = 2]$).

j-th column of $\mathbf{P}$, i.e. the area under the ROC curve by regarding p_{ij} $(y_i = \lambda_j)$ and $p_{i'j}$ $(y_{i'} = \lambda_k)$ as the numerical outputs on positive and negative classes respectively [13]. The following class-imbalance aware metrics are utilized for performance evaluation by aggregating across all class labels:

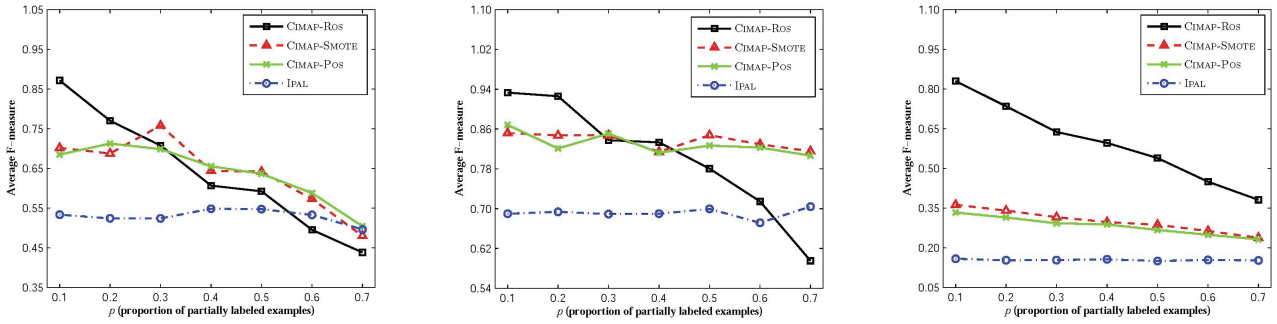
- `Average Precision`: $AvgP = \frac{1}{q}\sum_{j=1}^{q} P_j$
- `Average Recall`: $AvgR = \frac{1}{q}\sum_{j=1}^{q} R_j$
- `Average F-measure`: $AvgF = \frac{1}{q}\sum_{j=1}^{q} F_j$
- `MAUC`: $MAUC = \frac{2}{q(q-1)}\sum_{1\le j<k\le q}\frac{A_{jk}+A_{kj}}{2}$

For each of these metrics, it is obvious that the larger the metric value the better the classifier's performance.

4.1.3 Evaluation Protocol. As shown in Table 1, as a data-level approach for addressing the class-imbalance problem, CIMAP can be coupled with any off-the-shelf PL learning algorithm for model induction. In this paper, to show the effectiveness of CIMAP in endowing PL learning algorithm with the ability to yield class-imbalance aware classification model, four well-established methods are employed as the coupling algorithms including the k-nearest neighbor algorithm PL-KNN [16], the maximum margin algorithm PL-SVM [21], the convex optimization algorithm CLPL [9], and the instance-based algorithm IPAL [29]. Each coupling PL learning algorithm is instantiated with the parameter configuration suggested in respective literature. Furthermore, the two parameters k and τ for CIMAP are both fixed to be 5 in this paper.

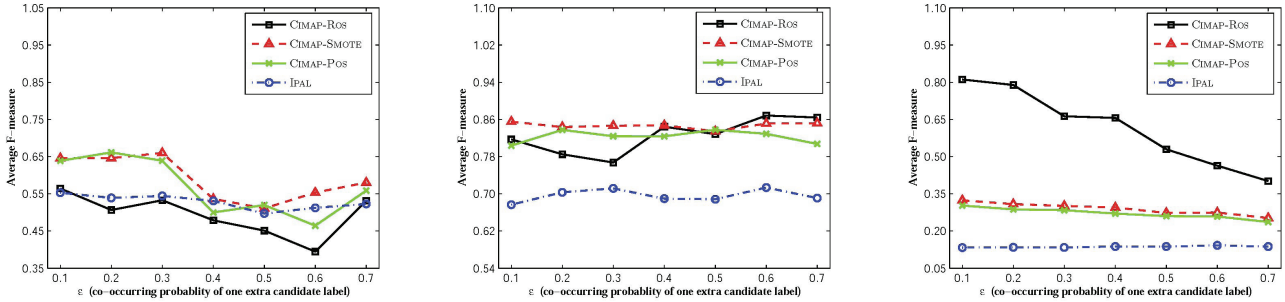
For each PL learning algorithm $\mathcal{L}$ ($\mathcal{L} \in \{$PL-KNN, PL-SVM, CLPL, IPAL$\}$), its performance is compared against the three variants of CIMAP coupled with $\mathcal{L}$. Without loss of generality, the three coupling variants are named as CIMAP-ROS, CIMAP-SMOTE, and CIMAP-POS respectively without explicitly referring to the coupling algorithm for notational brevity.

Ten-fold cross-validation is performed on each data set, where the mean metric value as well as standard deviation are recorded for each PL learning algorithm and the CIMAP variants. Next, detailed experimental results on artificial as well as real-world PL data sets are reported successively.



(a) Glass (b) Ecoli (c) Abalone

Figure 3: Average F-measure of the coupling algorithm IPAL and its CIMAP variants changes as p (proportion of partially labeled examples) increases from 0.1 to 0.7 (with two false positive candidate label $[r = 3]$).



(a) Glass (b) Ecoli (c) Abalone

Figure 4: Average F-measure of the coupling algorithm IPAL and its CIMAP variants changes as ϵ (co-occurring probability of the extra label) increases from 0.1 to 0.7 (with 100% partially labeled examples $[p = 1]$ and one false positive candidate label $[r = 1]$).

4.2 Controlled UCI Data Sets

In terms of *average F-measure*, Figures 1 to 3 illustrate the performance of IPAL and its CIMAP variants as p increases from 0.1 to 0.7 with step-size 0.1 ($r = 1, 2, 3$). Along with the ground-truth label, r class label(s) in $\mathcal{Y}$ will be randomly picked up to constitute the candidate label set. Similarly, Figure 4 illustrates the performance of IPAL and its CIMAP variants as ϵ increases from 0.1 to 0.7 with step-size 0.1 ($p = 1, r = 1$). Given any label $\lambda \in \mathcal{Y}$, one extra label $\lambda' \in \mathcal{Y}$ is designated to co-occur with λ in the candidate label set with probability ϵ. For brevity, figures for other coupling algorithms and evaluation metrics are not illustrated here while similar results can be observed as well.

As illustrated in Figures 1 to 4, in most cases the performance of CIMAP variants perform favorably against the original coupling PL learning algorithm. Furthermore, pairwise t-test at 0.05 significance level is conducted based on the results of ten-fold cross-validation. Table 4 reports the win/tie/loss counts on the coupling PL learning algorithm and its CIMAP variants in terms of each class-imbalance aware metric. For each coupling algorithm, out of the 1,008 statistical tests (28 configurations x 3 UCI data sets x 3 CIMAP variants x 4 evaluation metrics), it is shown that:

- For PL-SVM, all three CIMAP variants achieve superior or at least comparable performance than the coupling algorithm in terms of each evaluation metric;
- For IPAL, CIMAP-POS achieves superior or at least comparable performance across all evaluation metrics while CIMAP-ROS and CIMAP-SMOTE significantly outperforms the coupling algorithm in 61.9% and 88.0% cases respectively;
- For PL-KNN and CLPL, the three CIMAP variants are outperformed by the two coupling algorithms in only 2.2% and 1.6% cases respectively across all evaluation metrics.

4.3 Real-World Data Sets

Tables 5 and 6 report the detailed performance (mean±std) of each coupling PL learning algorithm and its CIMAP variants in terms of the four evaluation metrics. Pairwise t-test

Table 5: Detailed performance (mean±std) on the real-world PL data sets in terms of *average precision* and *average recall*. In addition, •/○ indicates whether the CIMAP variant is significantly superior/inferior to the coupling PL learning algorithm on each data set (pairwise t-test at 0.05 significance level).

PL algorithm and its CIMAP variants	Average precision					
	FG-NET	Lost	MSRCv2	BirdSong	Soccer Player	Yahoo! News
PL-KNN	0.017±0.011	0.408±0.076	0.338±0.064	0.511±0.034	0.277±0.019	0.647±0.022
CIMAP-ROS	0.219±0.027•	0.305±0.041○	0.513±0.031•	0.501±0.038	0.855±0.003•	0.783±0.003•
CIMAP-SMOTE	0.155±0.015•	0.450±0.023	0.396±0.029	0.481±0.019○	0.646±0.005•	0.790±0.002•
CIMAP-POS	0.114±0.011•	0.354±0.022	0.400±0.029•	0.456±0.022○	0.424±0.007•	0.693±0.004•
PL-SVM	0.026±0.013	0.637±0.074	0.308±0.057	0.606±0.049	0.366±0.014	0.685±0.024
CIMAP-ROS	0.101±0.030•	0.765±0.043•	0.536±0.057•	0.677±0.053	0.864±0.006•	0.843±0.003•
CIMAP-SMOTE	0.076±0.022•	0.644±0.037	0.322±0.030	0.593±0.029	0.521±0.006•	0.764±0.005•
CIMAP-POS	0.072±0.017•	0.531±0.037○	0.297±0.042	0.533±0.031○	0.423±0.007•	0.716±0.004•
CLPL	0.024±0.020	0.593±0.095	0.204±0.036	0.510±0.033	0.026±0.006	0.637±0.035
CIMAP-ROS	0.083±0.011•	0.711±0.032•	0.341±0.027•	0.551±0.010•	0.104±0.006•	0.736±0.009•
CIMAP-SMOTE	0.073±0.007•	0.625±0.044	0.252±0.038•	0.498±0.006	0.095±0.014•	0.649±0.055
CIMAP-POS	0.078±0.009•	0.520±0.028	0.258±0.038•	0.526±0.017	0.078±0.005•	0.639±0.041
IPAL	0.035±0.019	0.654±0.085	0.454±0.067	0.650±0.023	0.363±0.012	0.844±0.015
CIMAP-ROS	0.110±0.022•	0.518±0.049○	0.572±0.016•	0.643±0.009	0.869±0.004•	0.866±0.002•
CIMAP-SMOTE	0.102±0.009•	0.622±0.039	0.479±0.026	0.672±0.011•	0.678±0.004•	0.853±0.003
CIMAP-POS	0.100±0.017•	0.553±0.031○	0.487±0.030	0.521±0.015○	0.537±0.005•	0.757±0.005○
PL algorithm and its CIMAP variants	**Average recall**					
	FG-NET	Lost	MSRCv2	BirdSong	Soccer Player	Yahoo! News
PL-KNN	0.021±0.012	0.299±0.051	0.313±0.042	0.448±0.020	0.182±0.013	0.718±0.020
CIMAP-ROS	0.174±0.023•	0.296±0.024	0.364±0.018•	0.430±0.009	0.652±0.006•	0.783±0.003○
CIMAP-SMOTE	0.122±0.008•	0.379±0.013•	0.298±0.013	0.431±0.011	0.625±0.005•	0.731±0.002
CIMAP-POS	0.089±0.005•	0.340±0.019	0.299±0.010	0.402±0.008	0.397±0.004•	0.644±0.003○
PL-SVM	0.030±0.015	0.642±0.064	0.293±0.035	0.576±0.032	0.365±0.014	0.746±0.021
CIMAP-ROS	0.095±0.030•	0.734±0.045•	0.468±0.045•	0.705±0.046•	0.857±0.005•	0.873±0.003•
CIMAP-SMOTE	0.089±0.020•	0.641±0.032	0.303±0.020	0.618±0.028•	0.533±0.007•	0.839±0.003•
CIMAP-POS	0.078±0.011•	0.552±0.028○	0.288±0.023	0.574±0.022	0.418±0.007•	0.814±0.004•
CLPL	0.033±0.017	0.589±0.084	0.231±0.028	0.496±0.015	0.038±0.006	0.591±0.019
CIMAP-ROS	0.075±0.011•	0.687±0.022•	0.578±0.019•	0.338±0.017•	0.105±0.003•	0.839±0.003•
CIMAP-SMOTE	0.075±0.008•	0.631±0.028	0.272±0.019•	0.545±0.008•	0.099±0.004•	0.595±0.011
CIMAP-POS	0.084±0.010•	0.552±0.020	0.274±0.012•	0.564±0.017•	0.078±0.002•	0.594±0.031
IPAL	0.030±0.011	0.589±0.054	0.402±0.045	0.626±0.018	0.356±0.014	0.788±0.022
CIMAP-ROS	0.207±0.023•	0.508±0.031○	0.640±0.016•	0.635±0.007	0.847±0.004•	0.812±0.004•
CIMAP-SMOTE	0.091±0.014•	0.615±0.038	0.419±0.014	0.660±0.016•	0.667±0.004•	0.821±0.004•
CIMAP-POS	0.087±0.014•	0.549±0.031	0.423±0.016	0.513±0.013	0.519±0.006•	0.766±0.004○

at 0.05 significance level is conducted based on the ten-fold cross-validation, where the test outcomes between the coupling algorithm and its CIMAP variants are also recorded. Accordingly, it is impressive to observe that:

- On the `FG-NET` data set, all three CIMAP variants significantly outperform the coupling PL learning algorithm in all cases;
- On the `Soccer Player` data set, except that CIMAP-ROS is comparable to IPAL in terms of AUC, the three CIMAP variants significantly outperform the coupling algorithm in all the other cases;
- On the `MSRCv2` data set, out of 16 statistical tests (4 coupling algorithms x 4 evaluation metrics), CIMAP-ROS, CIMAP-SMOTE and CIMAP-POS perform significantly better in 93.7%, 43.7% and 50.0% cases respectively and perform comparably in the rest cases;
- On the `Lost`, `BirdSong`, and `Yahoo! News` data sets, out of 48 statistical tests (4 coupling algorithms x 4 evaluation metrics x 3 data sets), CIMAP-ROS, CIMAP-SMOTE and CIMAP-POS achieve superior or at least comparable performance in 79.1%, 93.8% and 66.6% cases respectively.

Table 6: Detailed performance (mean±std) on the real-world PL data sets in terms of *average F-measure* and *MAUC*. In addition, •/○ indicates whether the CIMAP variant is significantly superior/inferior to the coupling PL learning algorithm on each data set (pairwise t-test at 0.05 significance level).

PL algorithm and its CIMAP variants	Average F-measure					
	FG-NET	Lost	MSRCv2	BirdSong	Soccer Player	Yahoo! News
PL-KNN	0.014±0.008	0.302±0.051	0.294±0.042	0.445±0.023	0.186±0.012	0.661±0.015
CIMAP-ROS	0.173±0.022•	0.226±0.026○	0.359±0.021•	0.383±0.012○	0.824±0.004•	0.691±0.005
CIMAP-SMOTE	0.114±0.006•	0.349±0.016•	0.304±0.016	0.401±0.012○	0.622±0.004•	0.750±0.002•
CIMAP-POS	0.082±0.003•	0.289±0.016	0.307±0.011	0.385±0.011○	0.397±0.005•	0.656±0.003
PL-SVM	0.024±0.012	0.625±0.065	0.274±0.037	0.560±0.031	0.342±0.011	0.698±0.018
CIMAP-ROS	0.078±0.020•	0.724±0.047•	0.466±0.053•	0.677±0.052•	0.855±0.006•	0.857±0.003•
CIMAP-SMOTE	0.066±0.015•	0.621±0.029	0.283±0.024	0.579±0.032	0.519±0.007•	0.792±0.005•
CIMAP-POS	0.055±0.012•	0.519±0.025○	0.266±0.026	0.517±0.031○	0.409±0.007•	0.750±0.003•
CLPL	0.023±0.013	0.566±0.081	0.196±0.023	0.471±0.013	0.023±0.005	0.551±0.021
CIMAP-ROS	0.050±0.005•	0.672±0.024•	0.299±0.018•	0.552±0.011•	0.074±0.003•	0.764±0.005•
CIMAP-SMOTE	0.060±0.004•	0.604±0.029	0.225±0.017•	0.502±0.008•	0.062±0.003•	0.558±0.022
CIMAP-POS	0.066±0.004•	0.508±0.018	0.226±0.012•	0.516±0.019•	0.050±0.003•	0.556±0.028
IPAL	0.028±0.013	0.597±0.059	0.394±0.050	0.613±0.019	0.334±0.011	0.804±0.016
CIMAP-ROS	0.115±0.017•	0.449±0.026○	0.555±0.011•	0.576±0.006○	0.839±0.004•	0.827±0.003•
CIMAP-SMOTE	0.084±0.008•	0.595±0.038	0.430±0.016	0.652±0.013•	0.661±0.003•	0.827±0.003•
CIMAP-POS	0.079±0.012•	0.523±0.031○	0.433±0.016	0.545±0.013○	0.516±0.005•	0.753±0.004○
PL algorithm and its CIMAP variants	**MAUC**					
	FG-NET	Lost	MSRCv2	BirdSong	Soccer Player	Yahoo! News
PL-KNN	0.100±0.021	0.474±0.100	0.416±0.068	0.703±0.048	0.534±0.024	0.993±0.001
CIMAP-ROS	0.400±0.041•	0.652±0.062•	0.559±0.033•	0.692±0.062	0.999±0.001•	0.994±0.001•
CIMAP-SMOTE	0.578±0.023•	0.906±0.073•	0.899±0.069•	0.758±0.073	0.999±0.001•	0.993±0.001
CIMAP-POS	0.540±0.022•	0.906±0.072•	0.933±0.047•	0.970±0.002•	0.998±0.001•	0.988±0.001○
PL-SVM	0.197±0.042	0.804±0.086	0.422±0.085	0.718±0.099	0.816±0.031	0.990±0.004
CIMAP-ROS	0.456±0.040•	0.981±0.020•	0.751±0.146•	0.834±0.070•	0.994±0.004•	0.992±0.002
CIMAP-SMOTE	0.338±0.026•	0.983±0.010•	0.605±0.900•	0.867±0.065•	0.994±0.010•	0.987±0.001○
CIMAP-POS	0.378±0.045•	0.982±0.007•	0.556±0.066•	0.990±0.031•	0.987±0.006•	0.989±0.002
CLPL	0.197±0.042	0.804±0.086	0.422±0.085	0.718±0.099	0.816±0.031	0.990±0.004
CIMAP-ROS	0.456±0.040•	0.981±0.020•	0.751±0.146•	0.834±0.070•	0.994±0.004•	0.992±0.002•
CIMAP-SMOTE	0.338±0.026•	0.983±0.010•	0.605±0.090•	0.867±0.065•	0.994±0.010•	0.987±0.001
CIMAP-POS	0.378±0.045•	0.982±0.007•	0.556±0.066•	0.990±0.031•	0.998±0.001•	0.989±0.002
IPAL	0.297±0.053	0.717±0.152	0.774±0.102	0.922±0.081	0.875±0.030	0.999±0.001
CIMAP-ROS	0.392±0.037•	0.766±0.057	0.714±0.046	0.826±0.014○	0.880±0.002	0.883±0.003○
CIMAP-SMOTE	0.804±0.027•	0.998±0.001•	0.999±0.001•	0.999±0.001•	0.994±0.001•	0.999±0.001•
CIMAP-POS	0.758±0.045•	0.997±0.001•	0.999±0.001•	0.999±0.001•	0.994±0.001•	0.999±0.001

5 CONCLUSION

In this paper, the first attempt towards class-imbalance aware partial label learning is conducted. Specifically, a data-level solution named CIMAP is proposed by adapting the over-sampling techniques. As the ground-truth label of each PL training example is not directly accessible, a disambiguated multi-class data set is firstly generated via k-nearest neighbor aggregation. After that, the original PL training set is replenished via three over-sampling strategies as guided by the disambiguation results. Extensive comparative studies clearly validate the effectiveness of the proposed approach.

ACKNOWLEDGMENTS

This work was supported by the National Key R&D Program of China (2018YFB1004300), the National Science Foundation of China (61573104), the Fundamental Research Funds for the Central Universities (2242018K40082), and partially supported by the Collaborative Innovation Center of Novel Software Technology and Industrialization.

REFERENCES

[1] L. Abdi and S. Hashemi. 2016. To combat multi-class imbalanced problems by means of over-sampling techniques. *IEEE Transactions on Knowledge and Data Engineering* 28, 1 (2016), 238–251.

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

[3] K. Bache and M. Lichman. 2013. UCI Machine Learning Repository. School of Information and Computer Sciences, University of California, Irvine.

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

[5] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. 2002. SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research* 16, 1 (2002), 321–357.

[6] C.-H. Chen, V. M. Patel, and R. Chellappa. in press. Learning from ambiguously labeled face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (in press).

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

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

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

[10] F. Fernández-Navarro, C. Hervás-Martínez, and P. A. Gutiérrez. 2011. A dynamic over-sampling procedure based on sensitivity for multi-class problems. *Pattern Recognition* 44, 8 (2011), 1821–1833.

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

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

[13] D. J. Hand and R. J. Till. 2001. A simple generalisation of the area under the ROC curve for multiple class classification problems. *Machine Learning* 45, 2 (2001), 171–186.

[14] H. He and E. A. Garcia. 2009. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering* 21, 9 (2009), 1263–1284.

[15] T. R. Hoens, Q. Qian, N. V. Chawla, and Z.-H. Zhou. 2012. Building decision trees for the multi-class imbalance problem. In *Lecture Notes in Artificial Intelligence 7301*, P.-N. Tan, S. Chawla, C. K. Ho, and J. Bailey (Eds.). Springer, Berlin, 122–134.

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

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

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

[19] M. Lin, K. Tang, and X. Yao. 2013. Dynamic sampling approach to training neural networks for multiclass imbalance classification. *IEEE Transactions on Neural Networks and Learning Systems* 24, 4 (2013), 647–660.

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

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

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

[23] A. Sen, Md. M. Islam, K. Murase, and X. Yao. 2016. Binarization with boosting and oversampling for multiclass classification. *IEEE Transactions on Cybernetics* 46, 5 (2016), 1078–1091.

[24] Y. Sun, M. S. Kamel, and Y. Wang. 2006. Boosting for learning multiple classes with imbalanced class distribution. In *Proceedings of the 6th IEEE International Conference on Data Mining*. Hong Kong, China, 592–602.

[25] S. Wang and X. Yao. 2012. Multiclass imbalance problems: Analysis and potential solutions. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 42, 4 (2012), 1119–1130.

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

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

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

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

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

[31] M.-L. Zhang, B.-B. Zhou, and X.-Y. 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.

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

[33] Z.-H. Zhou and M.-L. Zhang. 2017. Multi-label learning. In *Encyclopedia of Machine Learning and Data Mining, 2nd Edition*, C. Sammut and G. I. Webb (Eds.). Springer, Berlin.

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