Partial Label Learning with Batch Label Correction

Yan Yan,[1,2] **Yuhong Guo**[2]
[1]School of Computer Science and Engineering, Northwestern Polytechnical University, China
[2]School of Computer Science, Carleton University, Canada
yanyan.nwpu@gmail.com, yuhong.guo@carleton.ca

Abstract

Partial label (PL) learning tackles the problem where each training instance is associated with a set of candidate labels, among which only one is the true label. In this paper, we propose a simple but effective batch-based partial label learning algorithm named PL-BLC, which tackles the partial label learning problem with batch-wise label correction (BLC). PL-BLC dynamically corrects the label confidence matrix of each training batch based on the current prediction network, and adopts a MixUp data augmentation scheme to enhance the underlying true labels against the redundant noisy labels. In addition, it introduces a teacher model through a consistency cost to ensure the stability of the batch-based prediction network update. Extensive experiments are conducted on synthesized and real-world partial label learning datasets, while the proposed approach demonstrates the state-of-the-art performance for partial label learning.

Introduction

In partial label (PL) learning, each training instance is assigned a set of candidate labels, only one of which is valid. In some literatures, this learning paradigm is also termed as superset label learning (Liu and Dietterich 2012; Hüllermeier and Cheng 2015) or ambiguous label learning (Hüllermeier and Beringer 2006; Zeng et al. 2013). Since precisely annotating the ground-truth label of each instance is typically difficult and costly, the task of partial label learning naturally arises in various application domains, such as automatic face naming (Hüllermeier and Beringer 2006; Zeng et al. 2013), web mining (Luo and Orabona 2010), and ecoinformatics (Liu and Dietterich 2014).

As the ground-truth label for each instance in the PL training set is hidden among the ambiguous labels in the candidate annotation set, one intuitive strategy of partial label learning is performing disambiguation, i.e., trying to identify the ground-truth label from the candidate label set. Existing disambiguation-based PL approaches can be roughly grouped into two categories, the average-based disambiguation approaches and the identification-based approaches. The average-based disambiguation approaches treat each candidate label in an equal manner for model induction, and make the final prediction by averaging the modeling outputs over all candidate labels (Hüllermeier and Beringer 2006; Cour, Sapp, and Taskar 2011; Zhang, Zhou, and Liu 2016). The identification-based disambiguation approaches take the ground-truth labels as latent variables and try to identify the ground-truth labels by employing an iterative procedure to gradually update the confidence value over each candidate label (Jin and Ghahramani 2003; Zhang and Yu 2015; Tang and Zhang 2017).

As it is more suitable to consider the different relevance degrees of each candidate label, the identification-based approaches have recently started gaining more attention from the research community. One recent work (Xu, Lv, and Geng 2019) attempts to recover the generalized label distribution by exploiting the topological information extracted based on a widely-used smooth assumption, and then learn a prediction model by fitting the recovered generalized label distribution. This phrase-wise learning methodology however is prone to the false positives in the labels identified via the generalized label distribution. Another recent work in (Lei and An 2019) learns the label confidence values of candidate labels by exploiting a self-training strategy by minimizing the widely used squared loss between the model predictions and the learned label confidence matrix. Although this work achieves some reasonable results, its label confidence score estimation is error-prone, which can have a profound negative impact on the model prediction, especially when the candidate label set is large.

In this paper, we propose a novel Partial Label Learning with Batch Label Correction (PL-BLC) method, which dynamically corrects the label confidence values of candidate labels and exploits a mixup data enhancement scheme to boost the prediction model through training batches. For each training batch, PL-BLC first dynamically corrects the confidence values of the candidate labels on being true labels based on the outputs of the current prediction network. Then PL-BLC mixes the partial label training instances with the corrected label confidence matrix by adopting a MixUp procedure, which serves as a data enhancement method to improve the prediction model's robustness against the noisy labels. In addition, PL-BLC further adopts a self-ensembling method to construct a teacher model for prediction, which maintains the stability of the batch updated prediction network and consequently enhances its batch-wise label correction capacity. We conduct extensive experiments on real-world and synthesized PL datasets under the partial label learning setting. The empirical results show the proposed PL-BLC achieves the state-of-the-art PL performance.

Related Work

Partial label learning is a prevalent classification problem in many real-world domains, where the true label of each training instance is hidden in the given candidate label set but unknown to the learning algorithm. It has some connection with noise label learning but addresses different problems. Noisy label learning (Natarajan et al. 2013) learns from training instances with corrupted labels, where the ground-truth labels are replaced by symmetric or asymmetric noisy labels, while in partial label learning the ground-truth label coexists with noisy labels.

To learn from PL data, one intuitive strategy is to treat all the candidate labels equally, and then average the outputs of all the candidate labels for final prediction. Following this strategy, some instances-based algorithms (Hüllermeier and Beringer 2006; Gong et al. 2017) predict the label $\mathbf{y}$ of a test instance $\mathbf{x}$ by averaging the outputs of its neighbors, i.e., $\arg\max_{\mathbf{y}\in\mathcal{Y}} \sum_{\mathbf{x}_i\in N(\mathbf{x})} \mathbb{I}(\mathbf{y} \in S_i)$ where $N(\mathbf{x})$ denotes the neighbors of $\mathbf{x}$ and S_i denotes the candidate label set for instance $\mathbf{x}_i$. Besides the instance-based instantiation, averaging strategy can also be applied on discriminative parametric models (Cour, Sapp, and Taskar 2011; Zhang, Zhou, and Liu 2016) by differentiating the average modeling output over the candidate labels from that over non-candidate labels, i.e., max $(\sum_{i=1}^{m}(\frac{1}{|S_i|}\sum_{\mathbf{y}\in S_i} F(\mathbf{x}_i, \mathbf{y}; \theta) - \frac{1}{|\widehat{S}_i|}\sum_{\widehat{\mathbf{y}}\in \widehat{S}_i} F(\mathbf{x}_i, \widehat{\mathbf{y}}; \theta)))$ where $\widehat{S}_i$ denotes the set of non-candidate labels. The simple average-based strategy however fails to explore the difference among the candidate labels and often produces unsatisfactory performance.

To address the drawback of average-based strategy, the identification-based strategy naturally arises due to its effectiveness of handling the candidate labels with discrimination. Existing approaches following this strategy consider the ground-truth label as a latent variable, and assume some parametric model $F(\mathbf{x}, \mathbf{y}; \theta)$ where the ground-truth can be identified by $\arg\max_{\mathbf{y}\in S_i} F(\mathbf{x}_i, \mathbf{y}; \theta)$. Some conventional methods try to optimize the objective function based on the maximum likelihood criterion (Jin and Ghahramani 2003) or the maximum margin criterion (Nguyen and Caruana 2008). Recently, exploiting the topological information in the feature space to derive the confidence score of each candidate label gets increasing attention from the research community (Zhang and Yu 2015; Zhang, Zhou, and Liu 2016; Feng and An 2018). The work in (Xu, Lv, and Geng 2019) proposes to iteratively update the generalized label distributions by leveraging the topological information in the feature space based on a widely-used smooth assumption, i.e., similar instance should have the same label, and learn a multi-class prediction model by fitting a regularized multi-output regressor with the generalized label distributions. However, the extracted topological information may not always be effective for helping the model training in PL settings. This method is also prone to errors induced in the generalized label distributions. In another recent work (Lei and An 2019), a self-training strategy is adopted to refine the label confidence scores of the candidate labels with the maximum infinity norm regularization. It performs partial label learning over the refined label confidence scores by minimizing a squared prediction loss. Although this work produces competitive performance, it is prone to the estimation error of label confidence scores, which can profoundly impair the prediction model due to the inherent property of alternative optimization.

MixUp (Zhang et al. 2017), which demonstrates outstanding robustness against noisy labels without explicitly modeling it, is one most popular data augmentation approach since its introduction. During the past few years, MixUp has been adopted to address different tasks, including semi-supervised learning (Berthelot et al. 2019), domain adaption (Mao et al. 2019), and learning with noisy labels (Arazo et al. 2019). The proposed work in this paper is the first one that exploits MixUp for partial label learning.

The Proposed Approach

Given a partial label training set $D = \{(\mathbf{x}_i, S_i)\}_{i=1}^{n}$, where $\mathbf{x}_i \in \mathcal{X} = \mathbb{R}^d$ is the input feature vector for the i-th instance, $S_i \subseteq \mathcal{Y}$ denotes the candidate label set associated with $\mathbf{x}_i$, and $\mathcal{Y}$ denotes the multi-class label space. The key assumption of partial label learning is that the ground-truth label $\mathbf{y}_i$ for each instance $\mathbf{x}_i$ is hidden in its candidate label set, i.e., $\mathbf{y}_i \in S_i$, but unknown to the learning algorithm. The task of partial label learning is to induce a multi-class classification model $F : \mathcal{X} \mapsto \mathcal{Y}$ from D. We use $B = \{(\mathbf{x}_i, S_i)\}_{i=1}^{m}$ to denote a batch of PL training instances sampled from D. In this section, we present a novel batch-based partial label learning method, PL-BLC. It performs batch-wise label correction and prediction network training in an online learning fashion, to gradually boost each other with a sequence of randomly sampled training batches. Each batch update involves three major operation components: *label correction*, which prepares a corrected soft label matrix based on current prediction outputs; *data enhancement*, which uses MixUp to induce robust data from the given batch for training the predictor; and *teacher model based consistency regularization*, which adopts a teacher model to ensure the consistency of the prediction network through noisy batches We elaborate these components and the proposed approach in the following subsections.

Label Correction

Comparing to standard multi-class learning, the main difficulty of partial label learning is that the ground-truth label coexists with additive noisy labels in the candidate label set but is unknown to the learning algorithm. The main challenge lies in correcting the partial label vector towards the true label indicator vector. Given a batch $B = \{(\mathbf{x}_i, S_i)\}_{i=1}^{m}$, without further information, each label in the candidate set has equal probability to be the ground-truth label. Then the initial prior label confidence matrix over the batch B can be written as $Q \in [0,1]^{m \times L}$, where L is the number of classes, such that

$$Q_{ij} = \begin{cases} 1/|S_i|, & \text{if} \quad \mathbf{y}_j \in S_i \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

To correct such a candidate uniform label confidence matrix, we propose to encode the global statistical data information observed so far by incorporating the label prediction outputs from the current prediction network $F(\mathbf{x}, \theta)$, which is trained over the previous batches. We assume the prediction network $F(\mathbf{x}, \theta)$ produces a multi-class probability vector with a softmax function for a given instance $\mathbf{x}$. We then can produce a corrected label confidence matrix $P \in [0,1]^{m \times L}$ by taking a prior confidence weighted convex combination of a uniform distribution and the prediction outputs of F:

$$P = Q + (1 - Q) \circ F(X_B, \theta) \tag{2}$$

where $X_B = [\mathbf{x}_1, \cdots, \mathbf{x}_m]^\top \in \mathbb{R}^{m \times d}$ denotes the input matrix in the batch B, $F(X_B, \theta)$ produces the $m \times L$ prediction probability matrix over the m instances, and "$\circ$" denotes the Hadamard matrix product operator. The prior label confidence matrix Q not only contributes the initial label information encoded from the pre-given candidate label set, but also controls the relative amount of information to accept from the prediction network. Note, in the extreme case, when there is only one label (i.e., the true label) in the candidate set, such a label correction will maintain the true label while rejecting potential noise from the prediction.

To maintain a valid label distribution over each instance and remove noise outside of the candidate set of labels, we further rescale P into $\widetilde{P}$ by renormalizing each row of P:

$$\widetilde{P}_{ij} = \begin{cases} \dfrac{P_{ij}}{\sum_{\mathbf{y}_k \in S_i} P_{ik}} & \text{if} \quad \mathbf{y}_j \in S_i \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

We treat $\widetilde{P}$ as the corrected soft label matrix for batch B. By replacing each candidate label set S_i with $\widetilde{P}_i$, we can obtain a corrected PL training batch $\widetilde{B} = \{(\mathbf{x}_i, \widetilde{P}_i)\}_{i=1}^{m}$. With the progress of prediction network training, this corrected label matrix will be more aligned with the true label matrix and consequently help further boost the prediction network. By using such a simple label correction method to dynamically update the batch labels in the training process, one can gradually mitigate the negative impact of the label noise without directly modeling the noise label distribution.

Data Enhancement with MixUp

MixUp is a data augmentation technique developed in (Zhang et al. 2017), which demonstrates strong robustness against noisy labels. generates augmenting instances from sample pairs $(\mathbf{x}_i, \mathbf{x}_j)$ and their corresponding labels $(\mathbf{y}_i, \mathbf{y}_j)$ as follows:

$$\widehat{\mathbf{x}} = \lambda \mathbf{x}_i + (1 - \lambda)\mathbf{x}_j, \quad \widehat{\mathbf{y}} = \lambda \mathbf{y}_i + (1 - \lambda)\mathbf{y}_j \tag{4}$$

where λ is randomly sampled from a beta distribution $\mathrm{Beta}(\alpha, \beta)$, for $\alpha, \beta \in (0, \infty)$. Training a prediction model on data generated from MixUp can encourage the model to behave linearly between training samples, which reduces oscillations in high density regions.

We adopt the MixUp scheme to enhance our corrected training batch, aiming to further improve the robustness of the prediction network against noisy labels. Specifically, we produce a mixup batch $\widehat{B}$ by randomly mixing up pairs of instances from $\widetilde{B}$ as follows:

$$\widehat{B} = \mathrm{MixUp}_{\lambda \sim \mathrm{Beta}(\alpha,\beta)}(\widetilde{B}, \overline{B}) \tag{5}$$

where $\overline{B}$ contains the instances of $\widetilde{B}$ after random order shuffling. That is, we mix up each corresponding pairs of instances, $(\mathbf{x}_i, \widetilde{P}_i) \in \widetilde{B}$ and $(\mathbf{x}_{\bar{i}}, \widetilde{P}_{\bar{i}}) \in \overline{B}$:

$$\widehat{\mathbf{x}}_i = \lambda \mathbf{x}_i + (1 - \lambda)\mathbf{x}_{\bar{i}}, \quad \widehat{P}_i = \lambda \widetilde{P}_i + (1 - \lambda)\widetilde{P}_{\bar{i}} \tag{6}$$

where $\mathbf{x}_{\bar{i}}$ denotes the i-th instance from $\overline{B}$, which is originally the $\bar{i}$-th instance from $\widetilde{B}$ before order shuffling. This yields a mixup enhanced batch $\widehat{B}$ with the same size.

In light of the robustness of MixUp against the noisy label information, we expect a prediction network $F(\mathbf{x}, \theta)$ trained on the mixup enhanced PL data can yield improved performance. Specifically, we adopt a least squared classification loss on the enhanced batch $\widehat{B}$ for training F:

$$\min_{F} \quad \mathcal{L}_c(\widehat{B}, F) = \frac{1}{m} \sum_{i=1}^{m} \|F(\widehat{\mathbf{x}}_i, \theta) - \widehat{P}_i\|_2^2 \tag{7}$$

Teacher Model based Consistency Regularization

To effectively integrate the consistent knowledge learned through sequences of batches and avoid possible volatility caused by the noisy partial labels, we propose to deploy a mean teacher model $F(\mathbf{x}, \theta')$ to assist the prediction network $F(\mathbf{x}, \theta)$, which can be treated as a student network. Following the self-ensembling strategy in (Tarvainen and Valpola 2017), the teacher model needs no separate training, but rather takes a weighted average of the student model along the training sequences of batches. Specifically, after observing each batch and updating the student model's parameter θ, the teacher model's parameter θ' can be updated as the exponential moving average (EMA) of the student model:

$$\theta' = \gamma \theta' + (1 - \gamma)\theta \tag{8}$$

where γ is a smoothing coefficient hyperparameter.

As the teacher model is less affected by the noisy labels in any single batch, we deploy the teacher model under a prediction consistency loss $\mathcal{L}_{c'}$ to assist the learning of the student prediction network F:

$$\mathcal{L}_{c'}(B, F) = \sum_{\mathbf{x} \sim B} \mathrm{KL}(F(\mathbf{x}, \theta), F(\mathbf{x}, \theta')) \tag{9}$$

where $\mathrm{KL}(\cdot, \cdot)$ is the Kullback-Leibler divergence between the softmax prediction outputs from the prediction network and the teacher model. We expect this prediction consistency loss can help enhance the label correction against noise and improve partial label learning.

Although the label correction step can shift a noisy candidate uniform label vector Q_i towards a label differentiating label vector $\widetilde{P}_i$, $\widetilde{P}_i$ still presents much higher entropy comparing to a ground-truth one-hot label vector. To promote the discriminativity of the prediction network over labels, we then propose to sharpen the prediction outputs of the teacher model and consequently impact the student model through the prediction consistency loss above. Let $\mathbf{q} = F(\mathbf{x}, \theta')$ denote the prediction probability vector of the teacher model over an input instance $\mathbf{x}$, we perform a common sharpening operation (LeCun, Bengio, and Hinton 2015) over $\mathbf{q}$:

$$\text{Sharpen}(\mathbf{q}, T)_i = \mathbf{q}_i^{\frac{1}{T}} \Big/ \sum_{j=1}^{L} \mathbf{q}_j^{\frac{1}{T}}, \quad \forall i \tag{10}$$

where T is the "temperature" hyperparameter that controls the degree of sharpening. Reducing the temperature encourages the model to produce low-entropy predictions.

Learning with PL-BLC

By integrating the classification loss in Eq.(7) and prediction consistency loss in Eq.(9) together, we get the overall batch-wise training loss for the proposed PL-BLC model:

$$\mathcal{L}(\widehat{B}, B, F) = \mathcal{L}_c(\widehat{B}, F) + \eta \mathcal{L}_{c'}(B, F) \tag{11}$$

where η is a trade-off hyperparameter that controls the relative importance of the classification loss and prediction consistency loss. We perform training to minimize this objective using a batch-based stochastic gradient descent algorithm. The full training algorithm is outlined in Algorithm 1.

Experiment

Datasets

We conducted controlled experiments on synthetic PL datasets constructed from 8 UCI datasets shown in Table 1. Following the widely-used controlling protocol (Wu and Zhang 2018; Xu, Lv, and Geng 2019; Lei and An 2019), we generate the synthetic PL datasets with three controlling parameters p, r and ϵ. Specifically, p controls the proportion of instances that have candidate labels (i.e., $S_i > 1$), r controls the number of noisy labels in the candidate label set (i.e., $|S_i| = r + 1$), and ϵ controls the probability of a specific noisy label co-occurring with the ground-truth label. For each UCI dataset, we generated multiple PL variants with different parameter configurations. We considered the following four groups of configurations: (I) $r = 1$, $p \in \{0.1, 0.2, \cdots, 0.7\}$; (II) $r = 2$, $p \in \{0.1, 0.2, \cdots, 0.7\}$; (III) $r = 3$, $p \in \{0.1, 0.2, \cdots, 0.7\}$; and (IV) $p = 1$, $r = 1$, $\epsilon \in \{0.1, 0.2, \cdots, 0.7\}$. Hence, in total we have 224 (28 config. × 8 datasets) generated synthetic PL datasets.

We also conducted experiments on six real-world PL datasets, which are collected from several application domains, including FG-NET (Panis and Lanitis 2014) for facial age estimation, Lost (Cour, Sapp, and Taskar 2011), Soccer Player (Zeng et al. 2013) and Yahoo! News (Guillaumin, Verbeek, and Schmid 2010) for automatic face naming from images or videos; MSRCv2 (Dietterich and Bakiri 1994) for object classification, and BirdSong (Briggs, Fern, and Raich 2012) for bird song classification. The characteristics of these datasets are reported in Table 2.

Algorithm 1 Training Algorithm of PL-BLC.

```
Input:
$D$ : the PL training set $\{(\mathbf{x}_i, S_i)\}_{i=1}^{n}$.
$F(\mathbf{x}, \theta)$: prediction neural network with parameters $\theta$.
$F(\mathbf{x}, \theta')$: teacher model with $\theta'$ equals to EMA of $\theta$.
$\gamma$: rate of EMA.
$\alpha, \beta$: Beta distribution parameters for $\text{MixUp}$.
$T$: sharpening temperature.
$\eta$: trade-off parameter.
for number of training iterations do
    Sample a batch $B$ with $m$ samples from $D$.
    Compute the prior label confidence matrix via Eq.(1).
    Compute the corrected label matrix $\widetilde{P}$ via Eq.(2), (3).
    $\overline{B} = \text{Shuffle}(\widetilde{B})$.
    Sample $\lambda$ from $\text{Beta}(\alpha, \beta)$.
    $\widehat{B} = \text{MixUp}_{\lambda}(\overline{B}, \widetilde{B})$.
    Compute the classification loss $\mathcal{L}_c(\widehat{B}, F)$ via Eq.(7).
    $\hat{q} = \text{Sharpen}(F(\mathbf{x}, \theta'), T)$
    $\mathcal{L}_{c'}(B, F) = \sum_{\mathbf{x} \sim B} \text{KL}(F(\mathbf{x}, \theta), \hat{q})$.
    $\mathcal{L}(\widehat{B}, B, F) = \mathcal{L}_c(\widehat{B}, F) + \eta \mathcal{L}_{c'}(B, F)$.
    Update the network parameter of $F$ by descending
        $\theta$ along it's stochastic gradient $\nabla_{\theta} \mathcal{L}(\widehat{B}, B, F)$.
    Update the teacher model $\theta' = \gamma\theta' + (1 - \gamma)\theta$.
end for
```

Table 1: Characteristics of the 8 UCI datasets.

Dataset	#Example	#Feature	#Class
glass	214	9	6
ecoli	336	7	8
deter	358	23	6
vehicle	846	18	4
segment	2310	18	7
satimage	6,345	36	7
usps	9,298	256	10
letter	20,000	16	26

Table 2: Characteristics of the real-world PL datasets.

Dataset	#Example	#Feature	#Class	avg.#CLs
FG-NET	1,002	262	78	7.48
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

Comparison Methods

We compared the proposed PL-BLC approach with the following PL methods, each configured with the suggested parameters according to the respective literature:

- PL-KNN (Hüllermeier and Beringer 2006): It uses k-NN method to learn from PL samples by weighed voting.
- PL-SVM (Nguyen and Caruana 2008): It uses SVM to learn from PL samples with l_2 regularization.
- CLPL (Cour, Sapp, and Taskar 2011): It decomposes the partial label learning problem into binary learning problems via feature mapping with convex loss optimization.
- PALOC (Wu and Zhang 2018): It uses a one-vs-one decomposition strategy to enable binary decomposition for leaning from PL samples.
- SURE (Lei and An 2019): it learns a confidence matrix of candidate labels with a self-training strategy and trains the prediction model over the learned label confidence matrix.

Implementation Details

We used a three-layer neural network as the prediction network for the proposed PL-BLC method. It uses the Leaky ReLu activation function in the two middle layers and uses softmax activation in the output layer. The two middle hidden layers have 512 and 256 hidden units respectively. The Adam (Kingma and Ba 2014) optimizer is adopted for training and the mini-batch size m is set to 32. We set α, β in Eq.(5) to 0.5 and 1 respectively. The learning rate, sharpening temperature T and the number of training iterations in Algorithm 1 are set to 0.0002, 0.4 and $100\times\lfloor n/32 \rfloor$ respectively. We selected the hyperparameter η from $\{0.001, 0.01, 0.1, 0.3, 0.5, 1, 10\}$ based on the classification loss value $\mathcal{L}_c$ in the training objective function; that is, the η value that leads to the smallest training $\mathcal{L}_c$ loss will be chosen. In terms of the EMA decay γ, we used γ = 0.99 during the ramp-up phase (for the first $20\times\lfloor n/32 \rfloor$ iterations in our experiment), and γ = 0.999 for the rest of training, since the student model improves quickly in the early phase. In the testing stage, we used the teacher model for prediction.

Results on Synthetic PL Datasets

On each PL dataset, we performed ten-fold cross-validation and report the average test accuracy results. First we study the comparison results under different groups of PL configurations. Figure 1 presents the results of all comparison methods for the configuration group I, where p increases from 0.1 to 0.7 with $r = 1$. In this setting, a candidate label set contains the ground-truth label and exactly one extra randomly chosen noisy label. Figure 2 presents the comparison results for the configuration group IV, where ϵ increases from 0.1 to 0.7 with $p = 1$ and $r = 1$. From both sets of figures, we can see that the proposed PL-BLC outperforms all the other comparison methods in most cases, which is not easy given the comparison methods have different strengths across different datasets. Especially, the performance gains yield by PL-BLC on the datasets of vehicle, segment, and satimage are quite remarkable under both settings I and IV. We obtained similar positive results for configuration group II and III. Due to the page limit, we do not include the figures but report the statistical results over all configurations below.

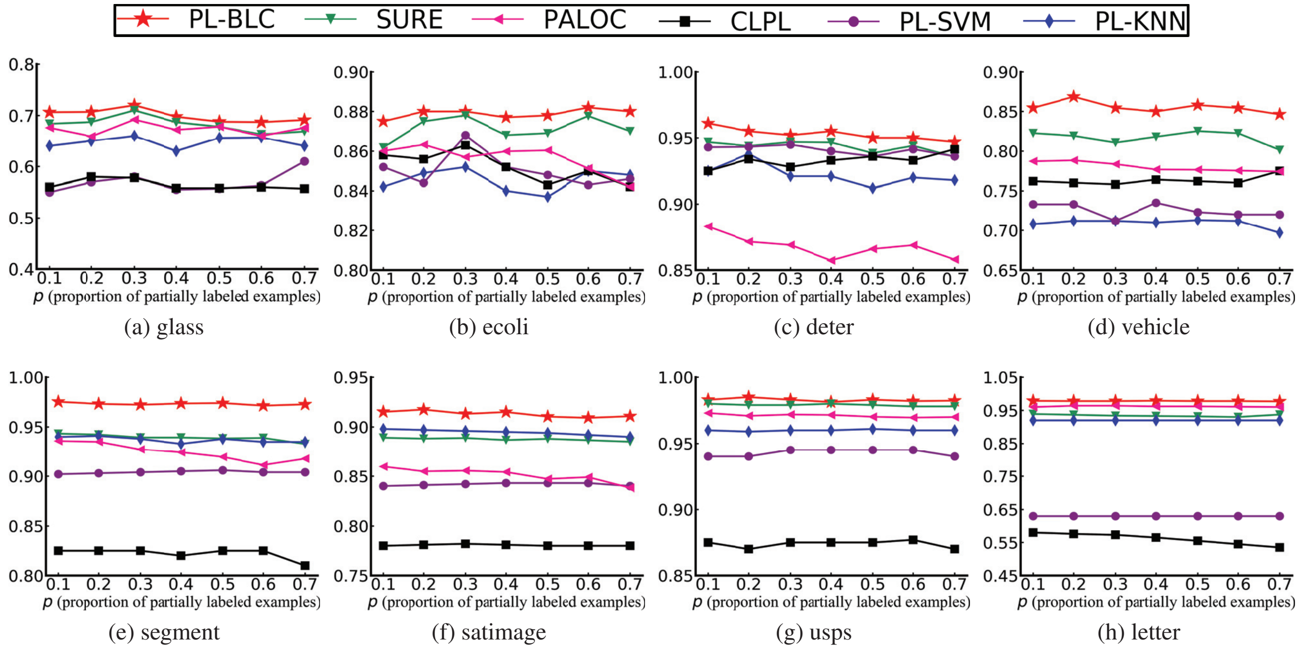
To statistically compare PL-BLC with the other comparison methods, we conducted pairwise t-test at 0.05 significance level on the ten-fold cross-validation results over all the 224 PL datasets obtained for all different configuration settings. The win/tie/loss counts between PL-BLC and each comparison method are reported in Table 3. From the 224 statistical tests, we can see that: 1) None of the comparison method outperform PL-BLC significantly in any controlled parameter configuration and on any UCI dataset. 2) Comparing to the averaging-based disambiguation methods, PL-BLC significantly outperforms PL-KNN, CLPL and PALOC in 69.6%, 72.7% and 66.9% of the cases respectively, and produces ties in the other cases. 3) Comparing to the identification-based disambiguation methods, PL-BLC significantly outperforms SURE and PL-SVM in 36.6% and 69.1% of cases and achieves comparable performance in the remaining 63.4% and 30.8% cases respectively. In summary, these results on the controlled UCI datasets clearly demonstrate the effectiveness of PL-BLC approach for partial label learning under different settings.

Results on Real-World PL Datasets

We conducted experiments on the six real-world PL datasets in a similar way with ten-fold cross-validation. Here we also compared to the results from the partial label learning method, PL-LE, in (Xu, Lv, and Geng 2019). Table 4 reports the mean test accuracy as well as the standard deviation results for all the comparison methods on these real-world PL datasets. In addition, we also conducted statistical pairwise t-test at 0.05 significance level based on the ten-fold cross-validation results, while the test outcomes between PL-BLC and the other comparison methods are recorded and indicated in Table 4 as well. From Table 4 we can observed that: 1) PL-BLC produces the best results on all the 6 datasets, and outperforms the other methods with remarkably performance gains. For example, PL-BLC outperforms the best alternative comparison methods by 3.7%, 2.6% and 2.6% on MSRCv2, Lost and Yahoo! News respectively. 2) Out of 30 cases (5 other comparison methods (except PL-LE) × 6 datasets), PL-BLC significantly outperforms all the comparison methods in 76.7% cases, and achieves competitive performance in 23.3% cases. 3) It is worth noting that PL-BLC is not significantly outperformed by any other comparison methods. These results on the real-world PL datasets again demonstrate the effectiveness of PL-BLC approach.

Ablation Study

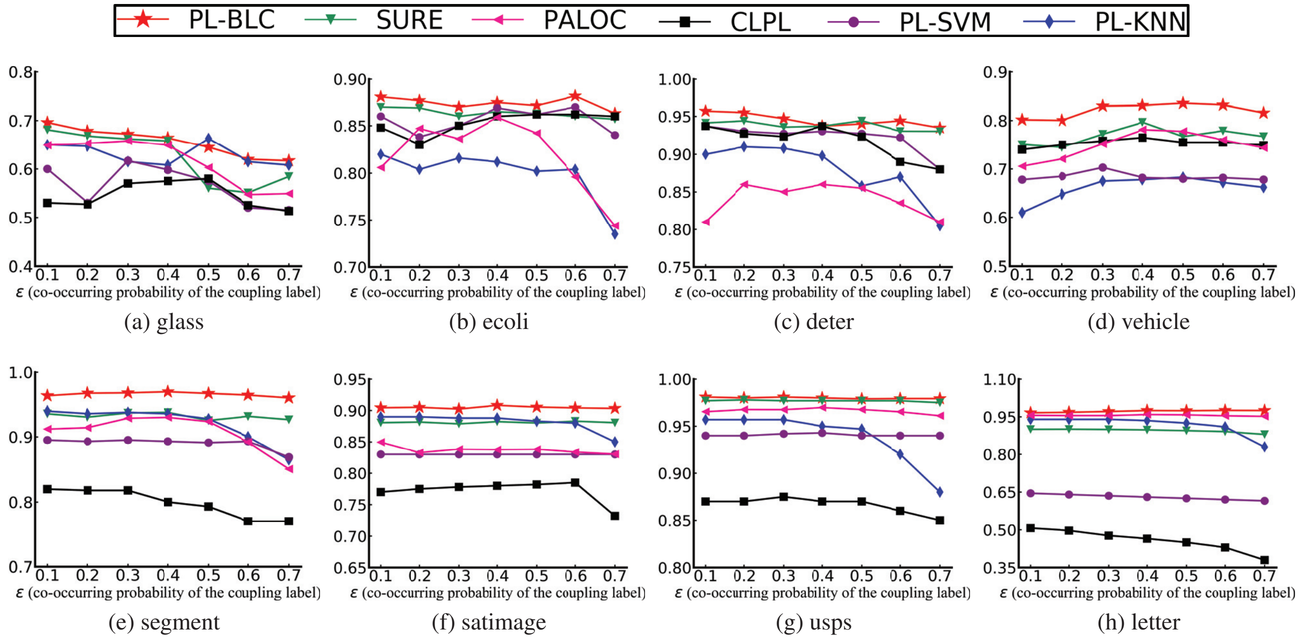
The proposed PL-BLC contains three components that contribute to PL learning: label correction, data enhancement with MixUp and teacher model based consistency regularization loss. To assess the importance of these components, we conduct an ablation study to compare PL-BLC with the following ablation variants: 1) CLS-MC, which drops the label correction; 2) CLS-LC, which drops the MixUp data enhancement; 3) CLS-LM, which drops the teacher model based prediction consistency loss; 4) CLS, which only uses the classification loss by dropping all the three components of label correction, data enhancement, and consistency regularization loss. It is a baseline variant. The comparison results are reported in Table 5. We can see that comparing to the full model, all four variants produced inferior results. Among the three variants, CLS-MC, CLS-LC and CLS-LM,



(a) glass (b) ecoli (c) deter (d) vehicle

(e) segment (f) satimage (g) usps (h) letter

Figure 1: Test accuracy of each comparison method 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) deter (d) vehicle

(e) segment (f) satimage (g) usps (h) letter

Figure 2: Test accuracy of each comparison method changes as ϵ (co-occurring probability of the coupling label) increases from 0.1 to 0.7 (with 100% partially labeled examples $[p = 1]$ and one false positive candidate label $[r = 1]$).

we observe that CLS-MC, which drops the label correction component, produces more inferior results, which suggests the label correction is more critical for the proposed model on addressing partial label learning. On the other hand, all the three variants, CLS-MC, CLS-LC and CLS-LM, outperform the baseline variant CLS across all the six real-world PL datasets, which suggests label correction, data enhancement and prediction consistency regularization all contribute to the proposed PL model. Overall, the ablation results suggest the proposed PL-BLC model is effective in integrating these components to address PL problems.

Table 3: Win/tie/loss counts of pairwise t-test (at 0.05 significance level) between PL-BLC and each comparison approach.

	PL-BLC vs –				
	SURE	PALOC	CLPL	PL-SVM	PL-KNN
varying p $[r = 1]$	22/34/0	40/16/0	42/14/0	42/14/0	41/15/0
varying p $[r = 2]$	20/36/0	36/20/0	40/16/0	39/17/0	38/18/0
varying p $[r = 3]$	18/38/0	36/20/0	40/16/0	36/20/0	38/18/0
varying ϵ $[p, r = 1]$	22/34/0	38/18/0	41/15/0	38/18/0	39/17/0
Total	**82/142/0**	**150/74/0**	**163/61/0**	**155/69/0**	**156/68/0**

Table 4: Test accuracy (mean±std) of each comparison method on the real-world PL datasets. ●/○ indicates whether PL-BLC is statistically superior/inferior to the comparison algorithm on each dataset (pairwise t-test at 0.05 significance level).

	FG-NET	Lost	MSRCv2	BirdSong	Soccer Player	Yahoo! News
PL-BLC	0.087±0.034	0.806±0.032	0.536±0.037	0.746±0.017	0.540±0.008	0.679±0.005
SURE	0.068±0.032	0.780±0.036	0.481±0.036●	0.728±0.024	0.533±0.017●	0.644±0.015●
PL-LE	0.082±0.023	0.773±0.043	0.499±0.037	0.730±0.013	0.536±0.020	0.653±0.006
PALOC	0.064±0.019	0.629±0.056●	0.479±0.042●	0.711±0.016●	0.537±0.015	0.625±0.005●
CLPL	0.063±0.027	0.742±0.038●	0.413±0.041●	0.632±0.019●	0.368±0.010●	0.462±0.009●
PL-SVM	0.063±0.029	0.729±0.042●	0.461±0.046●	0.660±0.037●	0.464±0.011●	0.629±0.010●
PL-KNN	0.038±0.025●	0.424±0.036●	0.448±0.037●	0.614±0.021●	0.497±0.015●	0.457±0.004●

Table 5: Comparison results of PL-BLC and its four ablation variants.

	FG-NET	Lost	MSRCv2	BirdSong	Soccer Player	Yahoo! News
PL-BLC	0.087±0.034	0.806±0.032	0.536±0.037	0.746±0.017	0.540±0.008	0.679±0.005
CLS-MC	0.065±0.029	0.616±0.038	0.442±0.057	0.637±0.015	0.481±0.012	0.529±0.013
CLS-LC	0.080±0.031	0.741±0.042	0.529±0.145	0.663±0.188	0.537±0.011	0.674±0.014
CLS-LM	0.076±0.025	0.773±0.033	0.506±0.045	0.706±0.017	0.536±0.010	0.658±0.008
CLS	0.060±0.030	0.562±0.042	0.414±0.059	0.620±0.032	0.470±0.009	0.525±0.012

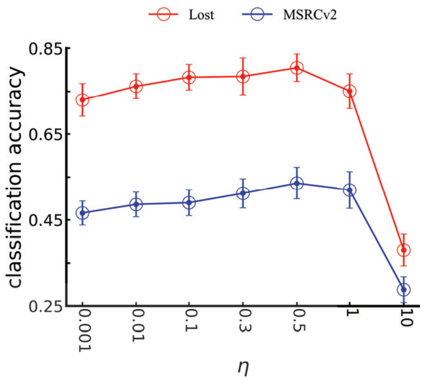


Figure 3: Parameter sensitivity analysis for PL-BLC on the Lost and MSRCv2 datasets.

Parameter Sensitivity Analysis

We also conducted parameter sensitivity analysis for the trade-off hyperparameter η of the proposed PL-BLC, which controls the weight of the prediction consistency loss, on two real-world PL datasets, Lost and MSRCv2. With the same experimental setting as above, we tested different η values from {0.001, 0.01, 0.1, 0.3, 0.5, 1, 10}. The test classification accuracy results (mean and standard deviations) for different η values are reported in Figure 3. We can see that when η is very small, the prediction consistency loss cannot contribute much. With the increase of η, the classification accuracy increases as the prediction consistency loss begins to contribute to the PL model, which suggests this regularization loss term is useful within a reasonable range of η values, e.g., $\eta \in [0.3, 0.5]$. However, when η is overly large (≥ 1), the performance degrades dramatically as the prediction consistency loss dominates. This is reasonable since the prediction consistency loss is expected to assist the prediction network, rather than dominate the learning.

Conclusion

In this paper, we proposed a novel batch-based label correction approach, LP-BLC, for partial label learning. Specifically, the proposed approach tackles partial label learning by dynamically updating the label confidence values of candidate labels following the identification-based strategy. Based on corrected label confidence values of the candidate labels, the proposed LP-BLC implements data enhancement by using MixUp which improves the model's robustness against irrelevant noisy labels. In addition, a teacher model is introduced to ensure the prediction network's stability with a consistency loss. The proposed approach is trained by batch-based stochastic gradient descent. Extensive experiments on synthesized and real-world datasets demonstrate that the proposed approach significantly outperforms the state-of-the-art partial label learning approaches.

Acknowledgments. This research was supported in part by the China Scholarship Council (201806290114), the NSERC discovery grant, and the Canada Research Chairs program.

References

Arazo, E.; Ortego, D.; Albert, P.; O'Connor, N. E.; and McGuinness, K. 2019. Unsupervised label noise modeling and loss correction. *arXiv preprint arXiv:1904.11238*.

Berthelot, D.; Carlini, N.; Goodfellow, I.; Papernot, N.; Oliver, A.; and Raffel, C. 2019. Mixmatch: A holistic approach to semi-supervised learning. *arXiv preprint arXiv:1905.02249*.

Briggs, F.; Fern, X. Z.; and Raich, R. 2012. Rank-loss support instance machines for miml instance annotation. In *the ACM SIGKDD international conference on Knowledge discovery and data mining*.

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

Dietterich, T. G., and Bakiri, G. 1994. Solving multiclass learning problems via error-correcting output codes. *Journal of artificial intelligence research* 2:263–286.

Feng, L., and An, B. 2018. Leveraging latent label distributions for partial label learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Gong, C.; Liu, T.; Tang, Y.; Yang, J.; Yang, J.; and Tao, D. 2017. A regularization approach for instance-based superset label learning. *IEEE transactions on cybernetics* 48(3):967–978.

Guillaumin, M.; Verbeek, J.; and Schmid, C. 2010. Multiple instance metric learning from automatically labeled bags of faces. In *European Conference on Computer Vision (ECCV)*.

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

Hüllermeier, E., and Cheng, W. 2015. Superset learning based on generalized loss minimization. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*.

Jin, R., and Ghahramani, Z. 2003. Learning with multiple labels. In *Advances in neural information processing systems (NeurIPS)*.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *nature* 521(7553):436.

Lei, F., and An, B. 2019. Partial label learning with self-guided retraining. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Liu, L., and Dietterich, T. G. 2012. A conditional multinomial mixture model for superset label learning. In *Advances in neural information processing systems (NeurIPS)*.

Liu, L., and Dietterich, T. 2014. Learnability of the superset label learning problem. In *International Conference on Machine Learning (ICML)*.

Luo, J., and Orabona, F. 2010. Learning from candidate labeling sets. In *Advances in neural information processing systems (NeurIPS)*.

Mao, X.; Ma, Y.; Yang, Z.; Chen, Y.; and Li, Q. 2019. Virtual mixup training for unsupervised domain adaptation. *arXiv preprint arXiv:1905.04215*.

Natarajan, N.; Dhillon, I. S.; Ravikumar, P. K.; and Tewari, A. 2013. Learning with noisy labels. In *Advances in neural information processing systems (NeurIPS)*.

Nguyen, N., and Caruana, R. 2008. Classification with partial labels. In *ACM SIGKDD international conference on Knowledge discovery and data mining*.

Panis, G., and Lanitis, A. 2014. An overview of research activities in facial age estimation using the fg-net aging database. In *European Conference on Computer Vision (ECCV)*.

Tang, C.-Z., and Zhang, M.-L. 2017. Confidence-rated discriminative partial label learning. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Tarvainen, A., and Valpola, H. 2017. Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. In *Advances in neural information processing systems (NeurIPS)*.

Wu, X., and Zhang, M.-L. 2018. Towards enabling binary decomposition for partial label learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Xu, N.; Lv, J.-Q.; and Geng, X. 2019. Partial label learning via label enhancement. In *AAAI Conference on Artificial Intelligence (AAAI)*.

Zeng, Z.; Xiao, S.; Jia, K.; Chan, T.-H.; Gao, S.; Xu, D.; and Ma, Y. 2013. Learning by associating ambiguously labeled images. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Zhang, M.-L., and Yu, F. 2015. Solving the partial label learning problem: An instance-based approach. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhang, M.-L.; Zhou, B.-B.; and Liu, X.-Y. 2016. Partial label learning via feature-aware disambiguation. In *ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.