Partial Label Learning via Label Influence Function

Xiuwen Gong [1] **Dong Yuan** [1] **Wei Bao** [1]

Abstract

To deal with ambiguities in partial label learning (PLL), state-of-the-art strategies implement disambiguations by identifying the ground-truth label directly from the candidate label set. However, these approaches usually take the label that incurs a minimal loss as the ground-truth label or use the weight to represent which label has a high likelihood to be the ground-truth label. Little work has been done to investigate from the perspective of how a candidate label changing a predictive model. In this paper, inspired by influence function, we develop a novel PLL framework called Partial Label Learning via Label Influence Function (PLL-IF). Moreover, we implement the framework with two specific representative models, an SVM model and a neural network model, which are called PLL-IF+SVM and PLL-IF+NN method respectively. Extensive experiments conducted on various datasets demonstrate the superiorities of the proposed methods in terms of prediction accuracy, which in turn validates the effectiveness of the proposed PLL-IF framework.

1. Introduction

In partial label learning (PLL) (Cour et al., 2011; Chen et al., 2014; Yu & Zhang, 2017), each instance is associated with a set of candidate labels, only one of which is the ground-truth label, while the others are false positive labels. This brings about ambiguities when training models for classification. In recent years, many real-world applications have arisen due to the growing demand for identifying the ground-truth label from partially labeled data. For example, in automatic face naming (Liu et al., 2016; Su et al., 2018), the learning system is required to recognize the name of each face from a candidate label set. In the image from a TV series (Fig.1 (a)) which contains ten people, each face is treated as an instance while names extracted from scripts constitute the candidate label set. Moreover, in automatic image annotation (Song et al., 2020; Chen et al., 2020), a web image (Fig.1 (b)) might be annotated online by different annotators with different labels, such as *cat*, *tiger* and *leopard*, forming the candidate label set, from which the ground-truth label is desired to be identified.

(a) Automatic face naming (b) Automatic image annotation

Figure 1: (a) Co-occurrence picture of the main characters in a TV series (i.e., Lost). Each face corresponds to one name in the candidate label set extracted from the script: **Sun, Hurley, Locke, Jack, Sayid, Kate, Claire, Sawyer, Charlie, and Boone**. (b) A web image is annotated by annotators online with different tags, such as, **cat**, **tiger** and **leopard**, but only one is the ground-truth label.

Partial label learning (PLL) aims to train a classifier from partially labelled data in order to automatically predict the ground-truth label for an unseen instance. The main challenge is that of how to deal with the ambiguities caused by false positive labels in candidate label set. The state-of-the-art strategy (Feng & An, 2019; Lyu et al., 2018; Xu et al., 2019) is to take the ground-truth label as a latent variable and then identify it directly from the candidate label set. The solution is usually obtained by employing an alternating algorithm to update the model papramter and the ground-truth label variable through an iterative refining procedure. On the one hand, when ground-truth label variable is fixed, the PLL problem turns out to be a well-studied multiclass optimization problem, and can be solved by any off-the-shelf multiclass implementations. On the other hand, when the model parameter is fixed, how to update the ground-truth label variable has become an intractable issue. As the binary ground-truth variable is discrete, the optimization is often NP-hard. Existing approaches usually take the label that

[1]Faculty of Engineering, The University of Sydney, NSW, Australia. Correspondence to: Xiuwen Gong <xiuwen.gong@sydney.edu.au>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

incurs a minimal loss as the ground-truth label or use the weight to represent which label has a high likelihood to be the ground-truth label. Little work has been done to investigate from the viewpoint of how a candidate label changing a predictive model, which could be applied as an indicator for identifying the ground-truth label.

Motivated by influence function (Hampel, 1974) that characterizes how a model's predictive loss changes when a small fraction of data points being perturbed, this paper first attempts to apply the similar idea to deal with PLL. However, due to the labeling ambiguity in the candidate label set, influence function cannot be applied to solve the PLL problem directly. In this paper, we adapt influence function to partial label learning and then propose a novel framework called Partial Label Learning via Label Influence Function (PLL-IF). Generally, the PLL-IF framework can be divided into two phases: 1) When the ground-labels are fixed, updating the model parameter; 2) When model parameter is fixed, updating the ground-truth label variables. In this work, we mainly focus on the second phase as the first phase could be easily solved by any off-the-shelf multiclass implementations given a specific loss function.

The main contributions of this work can be summarized as follows:

- We provide a new insight into partial label learning (PLL) from the perspective of influence function, and develop a novel framewok called Partial Label Learning via Label Influence Function (PLL-IF), which is generally formalized with an ERM-based optimizer.
- We define a quantity called Label Impact to quantify how a candidate label changes a predictive model, which can be further employed as an indicator to identify the most influential candidate label with highest impact on a model optimizer.
- We then introduce Label Influence Function (LIF) to efficiently approximate the label impact, which avoids retraining the model after each label is removed or perturbed, and largely reduces the heavy computation of the label impact.
- We further propose a novel ground-truth label identification method called Ground-truth Label Identification via Label Influence Function (GLI-LIF), which is the core part of the PLL-IF framework.
- We implement the PLL-IF framework through a representative non-neural network model (i.e., SVM model) and a basic neural network model, which we call PLL-IF+SVM method and PLL-IF+NN method respectively. Accordingly, we design two efficient optimization algorithms for the proposed two methods.
- We conduct extensive experiments on six synthetic datasets and six real-world datasets to validate the effectiveness of the proposed PLL-IF+SVM and PLL-IF+NN methods. The results demonstrate that the proposed methods outperform the state-of-the-art methods in terms of prediction accuracy, which in turn validates the effectiveness of the proposed PLL-IF framework.

2. Related Work

Partial label learning (PLL), also known as ambiguous-label learning (Chen et al., 2018), (Hüllermeier & Beringer, 2006), (Zeng et al., 2013) or superset-labellearning (Gong et al., 2018), (Liu & Dietterich, 2014), (Liu & Dietterich, 2012), is a weakly supervised learning problem (Zhou, 2017), which differs from semi-supervised learning (Belkin et al., 2006), (Berthelot et al., 2019). In PLL, each instance has a collection of candidate labels, with only one ground-truth label and the rest being false positive labels, resulting in ambiguity while training classification models. Existing disambiguation methods for partial label learning can be broadly divided into two categories (Zhang & Yu, 2015), (Lyu et al., 2021), (Lyu et al., 2020), (Zhou et al., 2017): disambiguation by candidate label averaging methods, or disambiguation by ground-truth label identifying methods. For the averaging-based methods (Cour et al., 2011; Hüllermeier & Beringer, 2006; Zhang & Yu, 2015; Gong et al., 2021a), all candidate labels of each instance are treated equally as the ground-truth label, and the prediction is made by averaging the modeling outputs. However, this kind of methods can be easily misled by false-positive labels in the candidate label set, and thus fail to generalize well in testing. For the identification-based methods (Jin & Ghahramani, 2002; Liu & Dietterich, 2012; Nguyen & Caruana, 2008; Yu & Zhang, 2017; Chai et al., 2020), the ground-truth label is regarded as a latent variable and identified through an iterative refining procedure. Moreover, SOTA research proposed some labeling confidence-based approaches. For example, Zhang et al. (2016) and Wang et al. (2019) proposed feature-aware disambiguation methods to generate different labeling confidences over candidate label set by utilizing the static and adaptive graph structure of feature space. Yan & Guo (2020) proposed a batch-based partial label learning algorithm named PL-BLC, which dynamically corrects the label confidence matrix of each training batch through taking the prior averaging label confidence and the outputs of current prediction network. Xu et al. (2019) developed the PL-LE approach that learns from partial label examples via label enhancement, after which the generalized label distributions are recovered by leveraging the topological information of the feature space. Feng & An (2019) developed a self-training-based approach named SURE, which jointly trains models and performs pseudo-labeling by introducing the maximum infinity norm regularization on the

modeling outputs in order to automatically differentiate the ground-truth label with high confidence. Gong et al. (2021b) proposed Discriminative Metric Learning for PartialLabel Learning (DML-PLL), which aims to learn a Mahanalobisdistance metric discriminatively while identifying the groundtruthlabel iteratively for PLL. Recently, some neural network related methods are developed for PLL. For example, Yao et al. (2020) proposed Deep Discriminative CNN (D^2CNN) with temporal ensembling, which employs the deep convolutional neural networks to improve the representation ability and entropy-based regularizer to enhance the discrimination ability. Lv et al. (2020) proposed a progressive identification method named PRODEN for approximately minimizing the proposed risk estimator, which updates the model and the identification of true labels in a seamless manner. Feng et al. (2020) proposed the first generation model of candidate label sets, and develop two novel PLL methods (i.e., RC and CC) that are guaranteed to be provably consistent.

Although the above-mentioned works have promoted the development of PLL, little research has been done from the perspective of that how a candidate label changes a predictive model. Inspired by influence function (Hampel, 1974) in robust statistics that can characterize how a model changes when a small fraction of data points being perturbed, which has been applied to machine learning in recent years. Koh & Liang (2017) applied the influence function to understand the prediction of a black-box model. Debruyne et al. (2008), Liu et al. (2014) and Christmann & Steinwart (2004) used the influence function for model selections and cross validations. Recently, Basu et al. (2021) investigated the influence function in deep learning with non-convex loss functions. In the next section, we adapt the influence function to solve the PLL problem.

3. The Proposed Approach

Let $\mathcal{S} = \{(\mathbf{x}_i, Y_i)\}_{i=1}^n$ denote the partial label dataset drawn i.i.d. n times from some unkown distribution P. For each training point, we have an instance $\mathbf{x}_i \in \mathbb{R}^d$ with d features and a corresponding candidate label set $Y_i \subseteq \mathcal{Y} = \{1, \cdots, q\}$. Let y_i denote the ground-truth label of $\mathbf{x}_i$, which is known residing in the corresponding candidate label set Y_i, i.e., $y_i \in Y_i$, but cannot be directly accessible in the training phase. Moreover, we use $f(\mathbf{x}_i; \mathbf{w})$ to denote the prediction model, where $\mathbf{w}$ is the model parameter. Let $\ell(y_i, f(\mathbf{x}_i; \mathbf{w}))$ be the loss function, and we fold the regularization term in ℓ for simplicity.

3.1. Problem Setup

In this subsection, we employ an ERM-based optimizer to formalize the PLL framework, which is named Partial Label Learning via Label Influence Function (PLL-IF). The formulation of PLL-IF can be expressed as follows:

$$\begin{aligned} &\min_{\mathbf{w}, y_i} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(\mathbf{x}_i; \mathbf{w})) \\ &\textit{where} \qquad y_i \in Y_i, \quad i \in \{1, \cdots, n\} \end{aligned} \tag{1}$$

In the above PLL-IF framework, partial label learning aims to learn the model parameter $\mathbf{w}$, and then makes prediction for an unseen instance. However, the ground-truth label in PLL is ambiguous, which leads to the difficulty in the training process. To deal with the ambiguities in PLL, one popular and effective strategy is to take the ground-truth label as a latent variable and then identify the ground-truth label directly from the candidate label set, which can be called the identification-based methods. The implementation of this kind of methods is usually fulfilled by employing an alternating algorithm to optimize the model parameter $\mathbf{w}$ and the ground-truth labels y alternatively in an iterative way. Specifically, when y is fixed, the objective function for optimizing the model parameter $\mathbf{w}$ turns out to be the following formulation:

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(\mathbf{x}_i; \mathbf{w})) \tag{2}$$

Eq. (2) turns out to be a well-studied multiclass optimization problem, and can be solved by any off-the-shelf multiclass solvers (Dhar et al., 2019), (Liu et al., 2017), (Hiranandani et al., 2019). For example, when the loss function is specified as the hinge loss, any implementation for solving multiclass SVM can be utilized to compute the model parameter $\mathbf{w}$. On the other hand, when the model parameter $\mathbf{w}$ is fixed, how to identify the ground-truth label y has become an intractable issue. The objective function to optimize the ground-truth label variable turns out to be as follows:

$$\begin{aligned} &\min_{y_i} \frac{1}{n} \sum_{i=1}^{n} \ell(y_i, f(\mathbf{x}_i; \mathbf{w})) \\ &\textit{where} \qquad y_i \in Y_i, \quad i \in \{1, \cdots, n\} \end{aligned} \tag{3}$$

However, the optimization is often NP-hard as the ground-truth variable is discrete. We attempt to solve this problem by the following subsections.

3.2. Label Impact

In order to quantify the impact of a label on a predictive model, we define Label Impact through the change of the predictive model optimizer. Specifically, we define the impact of the j-th label (the currently identified label) and the l-th label (any of the other candidate labels) on the current predictive model respectively as follows:

Definition 3.1 (j-th Label Impact)**.** Let $\mathbf{z} = (\mathbf{x}, \mathbf{y})$ be a training point, where $\mathbf{x} \in \mathbb{R}^d$ denotes the d-dimension instance; $\mathbf{y} \in \{0, 1\}^q$ denotes the corresponding q-dimension

label vector. Let Y denote the candidate label set of the instance $\mathbf{x}$. Assume that the j-th label is the currently identified label of the training point $\mathbf{z}$, while l denotes any of the other candidate labels, where $j, l \in Y$ and $l \neq j$, and $\mathbf{y}^j = 1$ and $\mathbf{y}^l = 0$. Let $\hat{\mathbf{w}}$ denote the current predictive model minimizer, where $\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{y}_i, f(\mathbf{x}_i; \mathbf{w}))$; $\hat{\mathbf{w}}_{-\mathbf{z}}$ denotes the new optimizer trained on the samples after removing $\mathbf{z}$ (equals to removing the j-th label). Then, for the currently identified label j, we have the j-th Label Impact defined as $\Delta\hat{\mathbf{w}}_j = \hat{\mathbf{w}}_{-\mathbf{z}} - \hat{\mathbf{w}}$.

Definition 3.2 (l-th Label Impact)**.** Let $\mathbf{z}_\delta = (\mathbf{x}, \mathbf{y}_\delta)$ denote a new training point by perturbing the traning point $\mathbf{z}$'s label vector $\mathbf{y}$, where $\mathbf{y}_\delta = \mathbf{y} + \delta$, and δ is a q-dimensional vector with j-th element being -1, l-th element being 1, while other elements being 0. Let $\hat{\mathbf{w}}$ denote the current predictive model minimizer; let $\hat{\mathbf{w}}_{-\mathbf{z},+\mathbf{z}_\delta}$ represent the new parameter trained on the samples with $\mathbf{z}_\delta$ in place of $\mathbf{z}$. Then, for any of the other candidate labels $l \in Y \backslash j$, we have the l-th Label Impact defined as $\Delta\hat{\mathbf{w}}_l = \hat{\mathbf{w}}_{-\mathbf{z},+\mathbf{z}_\delta} - \hat{\mathbf{w}}$.

To obtain the optimizer $\hat{\mathbf{w}}_{-\mathbf{z}}$ and $\hat{\mathbf{w}}_{-\mathbf{z},+\mathbf{z}_\delta}$ defined in the label impacts, an intuitive method is to retrain the model after each label of a training point is removed or perturbed. However, this can be prohibitively expensive in computation. In the following subsection, we will address this problem by introducing Label Influence Function (LIF), which provides an efficient way to approximate the label impact.

3.3. Label Influence Function

As $\hat{\mathbf{w}}_{-\mathbf{z}}$ is the minimizer trained on the samples after removing $\mathbf{z}$, which can also be taken as the empirical loss upweighted by some small amount ϵ, we then can reformulate $\hat{\mathbf{w}}_{-\mathbf{z}}$ as follows:

$$\hat{\mathbf{w}}_{\epsilon,-\mathbf{z}} \triangleq \arg\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{y}_i, f(\mathbf{x}_i; \mathbf{w})) + \epsilon\ell(\mathbf{y}, f(\mathbf{x}, \mathbf{w})) \tag{4}$$

We then have $\Delta\hat{\mathbf{w}}_j = \hat{\mathbf{w}}_{\epsilon,-\mathbf{z}} - \hat{\mathbf{w}}$. Similarly, we can reformulate the minizer $\hat{\mathbf{w}}_{-\mathbf{z},+\mathbf{z}_\delta}$ after moving a small mass ε from $\mathbf{z}$ onto $\mathbf{z}_\delta$ as follows:

$$\begin{aligned}\hat{\mathbf{w}}_{\varepsilon,-\mathbf{z},+\mathbf{z}_\delta} \triangleq \arg\min_{\mathbf{w}} \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{y}_i, f(\mathbf{x}_i; \mathbf{w})) - \varepsilon\ell(\mathbf{y}, f(\mathbf{x}, \mathbf{w})) \\ + \varepsilon\ell(\mathbf{y}_\delta, f(\mathbf{x}, \mathbf{w}))\end{aligned} \tag{5}$$

As a result, we have $\Delta\hat{\mathbf{w}}_l = \hat{\mathbf{w}}_{\varepsilon,-\mathbf{z},+\mathbf{z}_\delta} - \hat{\mathbf{w}}$.

In order to approximate $\Delta\hat{\mathbf{w}}_j$, we can derive that

$$\Delta\hat{\mathbf{w}}_j \approx \epsilon \cdot \frac{d(\hat{\mathbf{w}}_{\epsilon,-\mathbf{z}} - \hat{\mathbf{w}})}{d\epsilon}\Big|_{\epsilon=0} = \epsilon \cdot \frac{d\hat{\mathbf{w}}_{\epsilon,-\mathbf{z}}}{d\epsilon}\Big|_{\epsilon=0} \tag{6}$$

Similarly, we can approximate $\Delta\hat{\mathbf{w}}_l$ by

$$\Delta\hat{\mathbf{w}}_l \approx \varepsilon \cdot \frac{d(\hat{\mathbf{w}}_{\varepsilon,-\mathbf{z},+\mathbf{z}_\delta} - \hat{\mathbf{w}})}{d\varepsilon}\Big|_{\varepsilon=0} = \varepsilon \cdot \frac{d\hat{\mathbf{w}}_{\varepsilon,-\mathbf{z},+\mathbf{z}_\delta}}{d\varepsilon}\Big|_{\varepsilon=0} \tag{7}$$

The details of these derivations can be refered to the Appendix A. We formally define the Label Influence Function (LIF) as follows:

Definition 3.3 (Label Influence Function)**.** For any data point $\mathbf{z}$ upweighted on the empirical loss of a predictive model optimizer $\hat{\mathbf{w}}$ by a small amount ϵ, the label influence function can be defined as

$$\mathcal{I}(\mathbf{z}) \triangleq \frac{d\hat{\mathbf{w}}_{\epsilon,-\mathbf{z}}}{d\epsilon}\Big|_{\epsilon=0} = -H_{\hat{\mathbf{w}}}^{-1}\nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x}; \hat{\mathbf{w}})) \tag{8}$$

where $H_{\hat{\mathbf{w}}} = \frac{1}{n}\sum_{i=1}^{n} \nabla_{\mathbf{w}}^2 \ell(\mathbf{y}_i, f(\mathbf{x}_i; \hat{\mathbf{w}}))$ is the Hessian matrix and is positive definite by assumption.

We then can efficiently approximate the label impact after removing z (equals to removing the j-th label) without retraining the model by computing the following formulation as follows:

$$\Delta\hat{\mathbf{w}}_j = \epsilon\mathcal{I}(\mathbf{z}) \tag{9}$$

Here, $\epsilon = -\frac{1}{n}$, this is because the impact of removing $\mathbf{z}$ is equivalent to upweighting the empirical loss by $-\frac{1}{n}$ in Eq. (4) . The detailed derivation of Eq. (9) can be found in the Appendix A.

By further derivation and Dedifinition 3.3, we can get the following formulation:

$$\begin{aligned}&\frac{d\hat{\mathbf{w}}_{\varepsilon,-\mathbf{z},+\mathbf{z}_\delta}}{d\varepsilon}\Big|_{\varepsilon=0} \\ &= -H_{\hat{\mathbf{w}}}^{-1}\big(\nabla_{\mathbf{w}}\ell(\mathbf{y}_\delta, f(\mathbf{x}; \hat{\mathbf{w}})) - \nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x}; \hat{\mathbf{w}}))\big) \\ &= \mathcal{I}(\mathbf{z}_\delta) - \mathcal{I}(\mathbf{z})\end{aligned} \tag{10}$$

Finally, we can approximate the l-th label impact after removing data point $\mathbf{z}$ and adding the updated data point $\mathbf{z}_\delta$ on the emprical loss of the current predictive model optimizer $\hat{\mathbf{w}}$ via label influenc function as follows:

$$\Delta\hat{\mathbf{w}}_l = \varepsilon(\mathcal{I}(\mathbf{z}_\delta) - \mathcal{I}(\mathbf{z})) \tag{11}$$

Here, $\varepsilon = \frac{1}{n}$, as the impact of removing $\mathbf{z}$ and adding the updated point $\mathbf{z}_\delta$ is equal to upweighting the empirical loss by $\frac{1}{n}$ in Eq. (5). The detailed derivation of Eq. (11) can be found in the Appendix B.

To this end, we can calculate the j-th label impact (i.e., the currently identified ground-truth label impact) on the current predictive model optimizer $\hat{\mathbf{w}}$ via the label influence function by Eq. (9) efficiently, and also approximate the l-th label impact (i.e., any of the other candidate labels impact) by Eq. (11).

Table 1: Statistics of synthetic PLL datasets.

Datasets	#Instances	#Features	#Classes
Glass	214	9	7
Ecoli	336	7	8
Dermatology	366	33	6
Vehicle	846	18	4
Segmentation	2310	19	7
Satellite	6435	36	7

Algorithm 1 CG Algorithm

Goal: Find the optimal solution to Eq. (13) and (14).
Input: Dataset in last iteration $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$; data point $\mathbf{z} = (\mathbf{x}, \mathbf{y})$, $\mathbf{z}_\delta = (\mathbf{x}, \mathbf{y}_\delta)$; current model parameter $\hat{\mathbf{w}}$, ε.
Output: The optimal solution θ^*.

Initialize $\theta_0 = \mathbf{0}$, $r_0 = -v_{z_\delta}$, $p_0 = r_0$;
for $t = 0, 1, 2, \ldots$ **do**
  If $\|r_t\| \leq \varepsilon$, then $\theta^* = \theta_t$, terminate;
  $\alpha_t = \frac{(r_t)^T r_t}{(p_t)^T H_{\hat{W}} p_t}$;
  $\theta_{t+1} = \theta_t + \alpha_t p_t$;
  $r_{t+1} = r_t - \alpha_t H_{\hat{W}} p_t$;
  $\beta_t = \frac{r_{t+1}^T r_{t+1}}{r_t^T r_t}$;
  $p_t = r_{t+1} + \beta_t p_t$;
end for
Return the optimal solution $\theta^* = \theta_t$.

3.4. Ground-truth Label Identification via Label Influence Function

In this subsection, we propose a novel ground-truth label identification method called Ground-truth Label Identification via Label Influence Function (GLI-LIF), which is the core part of the PLL-IF framework.

Combining the label impact and label influence function, we can identify the label with highest impact on the current predictive model optimizer as the ground-truth label for any instance. Finally, we get the objective function for the proposed GLI-LIF as follows:

$$y = \arg \max_{j \in Y, l \in Y \backslash j} \left\{ \|\Delta \hat{\mathbf{w}}_j\|_1, \|\Delta \hat{\mathbf{w}}_l\|_1 \right\} \tag{12}$$

where Y is the candidate label set of instance $\mathbf{x}$; y is the ground-truth label. $\|\cdot\|_1$ denotes the ℓ_1-norm.

3.5. Optimization and Implementation Algorithms

There are some computation challenges in calculating the label influence function in Eq. (9) and (11) due to the forming and inverting of Hessian matrix $H_{\hat{\mathbf{w}}}^{-1}$. Therefore, we need to optimize the computation of influence function $\mathcal{I}(\mathbf{z}) = -H_{\hat{\mathbf{w}}}^{-1} v_z$, where $v_z = \nabla_{\mathbf{w}} \ell(\mathbf{y}, f(\mathbf{x}; \hat{\mathbf{w}}))$. We can use Hessian-vector products technique to implicitly approximate $H_{\hat{\mathbf{w}}}^{-1} v_z$ rather than directly computing $H_{\hat{\mathbf{w}}}^{-1}$. Following (Pearlmutter, 1994), we define $H_{\hat{\mathbf{w}}}^{-1} v_z \overset{def}{=} \theta_z^*$, which is transformed into a quadratic optimization problem as follows:

$$\theta_z^* = \arg \min_{\theta \in \mathbb{R}^d} \theta^T H_{\hat{\mathbf{w}}} \theta - v_z^T \theta \tag{13}$$

Similarly, we define $H_{\hat{\mathbf{w}}}^{-1} v_{z_\delta} \overset{def}{=} \theta_{z_\delta}^*$, thus we have

$$\theta_{z_\delta}^* = \arg \min_{\theta \in \mathbb{R}^d} \theta^T H_{\hat{\mathbf{w}}} \theta - v_{z_\delta}^T \theta \tag{14}$$

We then apply Conjugate Gradient (CG) to get the optimal solution for Eq. (13) and (14). The pseudo-code of CG method is presented in Algorithm 1.

In order to show that the proposed PLL-IF framework is compatible with a wide family of learning models, we implement PLL-IF by fitting two representative models into the proposed framework: One is the hinge loss-based SVM model; the other is the cross-entropy loss-based neural network model.

Algorithm 2 PLL-IF+SVM Algorithm

Goal: Minimize the optimization problem in Eq. (2).
Input: PLL training data $\mathcal{S}$, σ.
Output: The optimal solution $\mathbf{w}$ of Eq. (2).

Initialize ground-truth label y_i randomly chosen from Y_i for each instance $\mathbf{x}_i$; initialize the objective function value $\mathbf{F}^{(0)}$;
for $t = 1, 2, \ldots$ **do**
  Update the model parameter $\mathbf{w}$ by the LIBLINEAR SVM solver;
  Calculate the objective function value in Eq. (2), denoted as $\mathbf{F}^{(t)}$;
  if $|\frac{\mathbf{F}^{(t)} - \mathbf{F}^{(t-1)}}{\mathbf{F}^{(t-1)}}| \leq \sigma$ **then**
    terminate;
  end if
  for $i = 1, \ldots,$ n **do**
    $z_i = \{\mathbf{x}_i, y_i\}$;
    $j = y_i$;
    Calculate $\theta_{z_i}^*$ in Eq. (13) by Algorithm 1;
    Calculate $\Delta \hat{\mathbf{w}}_j$ according to Eq. (9);
    for $l \in Y_i \backslash j$ **do**
      $z_\delta = \{\mathbf{x}_i, l\}$;
      Calculate $\theta_{z_\delta}^*$ in Eq. (14) by Algorithm 1;
      Calculate $\Delta \hat{\mathbf{w}}_l$ according to Eq. (11);
    end for
    Update the ground-truth label y_i via Eq. (12);
  end for
  Construct new training dataset $\mathcal{S}^{(t)} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$;
end for

Table 2: Statistics of real-world PLL datasets.

Datasets	#Instances	#Features	#Classes	#Candidate Labels (avg.)	Domain
Lost	1122	108	16	2.23	automatic face naming from videos
MSRCv2	1758	48	23	3.16	object classification
Mirflickr	2780	1536	14	2.76	web image classification
BirdSong	4998	38	13	2.18	bird song classification
Soccer Player	17472	279	171	2.09	automatic face naming from images
Yahoo! News	22991	163	219	1.91	automatic face naming from images

Algorithm 3 PLL-IF+NN Algorithm

Goal: Minimize the optimization problem in Eq. (2).
Input: PLL training data $\mathcal{S}$; the number of epochs T.
Output: The optimal solution $\mathbf{w}$ of Eq. (2).

Let $\mathcal{A}$ be the Adam optimizer algorithm; Initialize ground-truth label y randomly chosen from Y for each instance;
for $t = 1$ to T **do**
  Shuffle training set into B mini-batches;
  for $k = 1, \ldots,$ B **do**
    Compute the objective function in Eq. (2), denoted as $\mathbf{F}^{(t)}$;
    Set gradient $-\nabla_{\mathbf{w}}\mathbf{F}^{(t)}$;
    Update the model parameter $\mathbf{w}$ by $\mathcal{A}$;
  end for
  for $i = 1, \ldots,$ n **do**
    $z_i = \{\mathbf{x}_i, y_i\}, j = y_i$;
    Calculate $\theta^*_{z_i}$ in Eq. (13) by Algorithm 1;
    Calculate $\Delta\hat{\mathbf{w}}_j$ according to Eq. (9);
    for $l \in Y_i \backslash j$ **do**
      $z_\delta = \{\mathbf{x}_i, y_i + \delta\} = \{\mathbf{x}_i, l\}$
      Calculate $\theta^*_{z_\delta}$ in Eq. (14) by Algorithm 1;
      Calculate $\Delta\hat{\mathbf{w}}_l$ according to Eq. (11);
    end for
    Update the ground-truth label y_i via Eq. (12);
  end for
  Construct new training dataset $\mathcal{S}^{(t)} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$;
end for

For the hinge loss-based SVM, we name the implementation of this method as PLL-IF+SVM. Specifically, we optimize the model parameter in the same way as that of M3PL (Yu & Zhang, 2017), and use the default setting in LIBLINEAR (Fan et al., 2008) with L2-regularized square hinge loss to train the classifier. However, we do not follow M3PL to optimize the ground-truth label variables. Instead, we adopt the proposed influence function-based method PLL-IF in Eq. (12) to optimize the ground-truth label variable y. Complete procedures to implement the proposed PLL-IF+SVM method can be found in Algorithm 2.

For the cross-entropy loss-based neural network, we name the implementation of this method as PLL-IF+NN. The implementation is based on PyTorch (Paszke et al., 2019). Specifically, we employ a 3-layer neural network for the proposed PLL-IF+NN, and use the Leaky ReLu activation function in the two middle layers with 512 and 256 hidden units respectively and employ softmax function in the output layer. The optimizer is Adam (Kingma & Ba, 2015) with the initial learning rate to be 0.0001. The mini-batch size is set to 32 and we train the model 500 epochs with cross-entropy loss and update ground-truth label variable every epoch. Complete procedures to implement the proposed PLL-IF+NN method can be found in Algorithm 3.

4. Experiments

In this section, we conduct experiments to evaluate the classification performance of the proposed PLL-IF and several state-of-the-art partial label learning methods in terms of classification accuracy.

4.1. Datasets and Baselines

We conducted controlled experiments on synthetic PLL datasets configured from six UCI datasets[1], which are summarized in Table 1. Given the general controlling strategy over multi-class datasets (Cour et al., 2011), (Chen et al., 2014), (Liu et al., 2019), synthetic partial-label datasets can be generated from controlled UCI datasets by configuring two controlling parameters p and r, where p controls the proportion of instances that have candidate labels (i.e.$|Y_i| > 1$), and r controls the number of false positive labels in the candidate label set (i.e., $r = |Y_i| - 1$). In practice, we consider configurations where p increases from 0.1 to 0.9 with step-size 0.2 and $r = 1, 2, 3$ for each UCI dataset. For each partial label instance, the candidate label set contains the ground-truth label along with r additional labels randomly chosen from all the classes. Hence, we have 162 (27 configurations $\times$ 6 datasets) generated synthetic PLL datasets.

We also conducted experiments on six real-world PLL datasets, which are summarized in Table 2. These datasets are collected from various domains, such as automatic face naming from videos including Lost (Cour et al.,

[1] http://archive.ics.uci.edu/ml

Table 3: Win/tie/loss counts of pairwise t-test (at 5% significance level) between PLL-IF+NN and each baseline.

Config. \ Method	PLL-IF+NN vs -					
	PL-SVM	CLPL	M3PL	SURE	PRODEN	PL-BLC
varying p, r = 1	54\0\0	54\0\0	54\0\0	53\1\0	53\1\0	47\7\0
varying p, r = 2	54\0\0	54\0\0	54\0\0	54\0\0	47\4\3	42\6\6
varying p, r = 3	54\0\0	54\0\0	54\0\0	54\0\0	50\3\1	46\3\5
Total	162\0\0	162\0\0	162\0\0	161\1\0	150\8\4	135\16\11

Table 4: Win/tie/loss counts of pairwise t-test (at 5% significance level) between PLL-IF+SVM and each baseline.

Config. \ Method	PLL-IF+SVM vs -					
	PL-SVM	CLPL	M3PL	SURE	PRODEN	PL-BLC
varying p, r = 1	54\0\0	51\3\0	45\5\4	40\4\10	24\4\26	9\3\42
varying p, r = 2	54\0\0	54\0\0	51\3\0	46\5\3	12\3\39	8\1\45
varying p, r = 3	54\0\0	52\2\0	49\4\1	38\4\12	8\3\43	16\5\33
Total	162\0\0	157\5\0	145\12\5	124\13\25	44\10\108	33\9\120

2011), object classification including MSRCv2 (Liu & Dietterich, 2012), web image classification including Mirflickr (Huiskes & Lew, 2008), bird song classification including BirdSong (Briggs et al., 2012), automatic face naming from images including Soccer Player (Zeng et al., 2013), and Yahoo! News (Guillaumin et al., 2010). The average number of candidate labels for each real-world partial label dataset are also recorded in Table 2.

We compare the proposed PLL-IF method with the following six state-of-the-art PLL approaches: *PL-SVM* (Nguyen & Caruana, 2008), *CLPL* (Cour et al., 2011), *M3PL* (Yu & Zhang, 2017), *SURE* (Feng & An, 2019), *PRODEN* (Lv et al., 2020), *PL-BLC* (Yan & Guo, 2020), where the last two are deep learning methods.

For each baseline, we follow the suggested configurations including the parameters and optimizers according to their respective literatures. For maximum margin based methods, such as, PL-SVM, M3PL methods, we only consider the linear versions for brevity. For the evaluation metric, we perform five-fold cross-validation on each dataset and report the mean accuracy with standard deviation of each method.

4.2. Performance Evaluation

1) Prediction Performance on Synthetic Datasets

Figs. 2 to 7 in Appendix C illustrate the performance comparisons of the proposed method (PLL-IF+NN / PLL-IF+SVM) with six state-of-the-art PLL baselines in terms of mean accuracy with varying proportion of partially labeled examples p on different datasets (i.e., Ecoli, Segmentation, Satellite, Glass, Dermatology, Vehicle). From trends in the figures, we make the following observations:

- The proposed PLL-IF+NN method consistently outperforms all baselines on all datasets with the increasing proportion of partially labeled examples (p), which is not easy considering that the comparison methods have their own strengths across different datasets.
- Additionally, the proposed PLL-IF+SVM is superior to most of the baselines, such as PL-SVM, CLPL, M3PL, SURE in most cases, while is comparable to PRODEN and PL-BLC on Glass, Ecoli, Dermatology, Satellite datasets when there are two false-positive labels (r=2); and we can obtain similar results on Segmentation (r=1), Vehicle (r=3) datasets.

We statistically compare the proposed PLL-IF+NN and PLL-IF+SVM with all baselines and conduct pairwise t-test at 5% significance level on five-fold cross-validation results over all the 162 synthetic PLL datasets obtained from different configuration settings. The win/tie/loss counts between PLL-IF and each each baseline are reported in Table 3 and 4 respectively. From the statistical test results, we can observe that:

- The proposed PLL-IF+NN significantly outperforms all baselines on all controlled parameter configurations. Concretely, PLL-IF+NN almost has no loss or tie compared with PL-SVM, CLPL, M3PL, SURE; while has few losses or ties compared with PRODEN and PL-BLC, but wins a lot more in total counts.
- The proposed PLL-IF+SVM outperforms PL-SVM, CLPL, M3PL, SURE on all configurations in the total number of wins. However, it underperforms PRODEN and PL-BLC with only one third counts of wins.

Overall, the proposed PLL-IF+NN approach is superior to all baselines on all controlled UCI datasets, while the proposed PLL-IF+SVM method is superior to most of the

Table 5: Mean accuracy ± standard deviation via five-fold cross validation on six real-world datasets for all methods. The best results are in bold. •/○ indicates that our method (PLL-IF+NN / PLL-IF+SVM) is significantly superior / inferior than the baseline (pairwise t-test at %5 significance level).

Method	Lost	MSRCv2	Mirflickr	BirdSong	Soccer Player	Yahoo!News
PLL-IF+NN	**0.809 ± .041**	**0.538 ± .027**	**0.569 ± .030**	**0.753 ± .003**	**0.560 ± .004**	**0.683 ± .007**
PLL-IF+SVM	0.782 ± .012	0.513 ± .022	0.561 ± .003	0.723 ± .017	0.554 ± .010	0.646 ± .026
PL-SVM	0.691 ± .012•	0.481 ± .037•	0.441 ± .061•	0.661 ± .067•	0.462 ± .006•	0.615 ± .015•
CLPL	0.732 ± .032•	0.433 ± .020•	0.549 ± .017	0.635 ± .019•	0.367 ± .004•	0.471 ± .049•
M3PL	0.747 ± .031•	0.499 ± .026•	0.480 ± .016•	0.694 ± .065•	0.440 ± .005•	0.623 ± .062•
SURE	0.767 ± .026	0.508 ± .021	0.562 ± .015	0.702 ± .025•	0.531 ± .014	0.632 ± .015•
PRODEN	0.765 ± .014	0.452 ± .017•	0.524 ± .011	0.721 ± .004	0.559 ± .005	0.674 ± .005
PL-BLC	0.806 ± .032	0.536 ± .037	0.558 ± .038	0.746 ± .017	0.540 ± .008	0.679 ± .005

baselines. These results clearly demonstrate the effectiveness of the proposed PLL-IF framework for partial label learning.

2) Prediction Performance on Real-world Datasets

We apply the proposed PLL-IF as well as six state-of-the-art PLL baselines on each real-world dataset to evaluate their effectiveness in terms of mean accuracy and standard deviation. In addition, we conduct the statistical pairwise t-test at the default 5% significance level to validate the superiority of the proposed method (PLL-IF+NN / PLL-IF+SVM) compared with the baselines. •/○ represents whether our method is significantly better/worse than the compared methods. The results are shown in Table 5. From the results, it is impressive to observe that:

- The proposed PLL-IF+NN consistently achieves the best results among all methods, inlcuding all baselines and the proposed PLL-IF+SVM on all the six real-world datasets, and outperforms non-nerual newtork-based methods like PL-SVM, CLPL, M3PL, SURE with remarkable improvements.

- The proposed PLL-IF+SVM is comparable or a little bit inferior to the latest neural network-based methods PRODEN and PL-BLC on most of the real-world datasets. However, as a non-nerual newtork-based method, PLL-IF+SVM performs much better than other non-nerual newtork-based methods, such as PL-SVM, CLPL, M3PL, SURE in most cases.

- Interestingly, we can see that the superiority of the neural network-based PLL-IF method (i.e., PLL-IF+NN) is much more obvious compared with both nerual newtork-based and non-nerual newtork-based methods, including the proposed PLL-IF+SVM across all datasets.

- Even though the proposed PLL-IF framework is loss-independent and classifier-independent, it seems more workable for neural network-based methods, despite of the superiorities of working with non-neural network-based methods.

- From the above observations, we hold that the proposed PLL-IF framework is effective for partial label learning, which in turn validates the effectiveness of employing the influence function as a strategy to update the ground-truth label variables.

These results on real-world datasets demonstrate the effectiveness of the proposed PLL-IF framework.

5. Conclusion

This work provides a novel insight into PLL from the perspective of influence function. Generally, we first formalize a general PLL framework called Partial Label Learning via Label Influence Function (PLL-IF) with an ERM-based optimizer. To identify the most influential candidate label with highest impact on a model optimizer, we define the Label Impact to quantify how a candidate label changes a predictive model. To avoid retraining the model repeatedly after each label is removed or perturbed in computing the label impact, we further introduce Label Influence Function (LIF) to efficiently approximate the label impact. Based on these, we develop a novel ground-truth label identification method called Ground-truth Label Identification via Label Influence Function (GLI-LIF), which is the core part of the PLL-IF framework. To illustrate the usage of the proposed framework, we implement the PLL-IF framwork through a representative non-neural network SVM model and a basic neural network model, which we call PLL-IF+SVM method and PLL-IF+NN method respectively. Accordingly, we design two efficient optimization algorithms for the proposed two methods. Lastly, we conduct extensive experiments on six synthetic datasets and six real-world datasets to demonstrate the effectiveness of the proposed methods, and the results show the superiorities of PLL-IF+SVM and PLL-IF+NN methods compared with the state-of-the-art methods in terms of prediction accuracy, which in turn validates the effectiveness of the proposed PLL-IF framework.

References

Basu, S., Pope, P., and Feizi, S. Influence functions in deep learning are fragile. In *ICLR*, 2021.

Belkin, M., Niyogi, P., and Sindhwani, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7:2399–2434, 2006.

Berthelot, D., Carlini, N., Goodfellow, I. J., Papernot, N., Oliver, A., and Raffel, C. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, pp. 5050–5060, 2019.

Briggs, F., Fern, X. Z., and Raich, R. Rank-loss support instance machines for MIML instance annotation. In *KDD*, pp. 534–542, 2012.

Chai, J., Tsang, I. W., and Chen, W. Large margin partial label machine. *IEEE Transactions on Neural Networks and Learning Systems*, 31(7):2594–2608, 2020.

Chen, C., Patel, V. M., and Chellappa, R. Learning from ambiguously labeled face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(7):1653–1667, 2018.

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

Chen, Y., Zeng, X., Chen, X., and Guo, W. A survey on automatic image annotation. *Applied Intelligence*, 50(10): 3412–3428, 2020.

Christmann, A. and Steinwart, I. On robustness properties of convex risk minimization methods for pattern recognition. *Journal of Machine Learning Research*, 5:1007–1034, 2004.

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

Debruyne, M., Hubert, M., and Suykens, J. A. Model selection in kernel based regression using the influence function. *Journal of Machine Learning Research*, 9:2377–2400, 2008.

Dhar, S., Cherkassky, V., and Shah, M. Multiclass learning from contradictions. In *NeurIPS*, pp. 8398–8408, 2019.

Fan, R., Chang, K., Hsieh, C., Wang, X., and Lin, C. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

Feng, L. and An, B. Partial label learning with self-guided retraining. In *AAAI*, pp. 3542–3549, 2019.

Feng, L., Lv, J., Han, B., Xu, M., Niu, G., Geng, X., An, B., and Sugiyama, M. Provably consistent partial-label learning. In *NeurIPS*, 2020.

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

Gong, X., Yang, J., Yuan, D., and Bao, W. Generalized large margin knn for partial label learning. *IEEE Transactions on Multimedia*, 2021a. doi: 10.1109/TMM.2021.3109438.

Gong, X., Yuan, D., and Bao, W. Discriminative metric learning for partial label learning. *IEEE Transactions on Neural Networks and Learning Systems*, 2021b. doi: 10.1109/TNNLS.2021.3118362.

Guillaumin, M., Verbeek, J. J., and Schmid, C. Multiple instance metric learning from automatically labeled bags of faces. In *ECCV*, pp. 634–647, 2010.

Hampel, F. R. The influence curve and its role in robust estimation. *Journal of the American Statistical Association*, 69(346):383–393, 1974.

Hiranandani, G., Boodaghians, S., Mehta, R., and Koyejo, O. Multiclass performance metric elicitation. In *NeurIPS*, pp. 9351–9360, 2019.

Huiskes, M. J. and Lew, M. S. The MIR flickr retrieval evaluation. In *SIGMM*, pp. 39–43, 2008.

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

Jin, R. and Ghahramani, Z. Learning with multiple labels. In *NeurIPS*, pp. 897–904, 2002.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Koh, P. W. and Liang, P. Understanding black-box predictions via influence functions. In *ICML*, pp. 1885–1894, 2017.

Liu, L. and Dietterich, T. G. Learnability of the superset label learning problem. In *ICML*, pp. 1629–1637, 2014.

Liu, L. P. and Dietterich, T. G. A conditional multinomial mixture model for superset label learning. In *NeurIPS*, pp. 557–565, 2012.

Liu, W., Tsang, I. W., and Müller, K. An easy-to-hard learning paradigm for multiple classes and multiple labels. *Journal of Machine Learning Research*, 18:94:1–94:38, 2017.

Liu, W., Xu, D., Tsang, I. W., and Zhang, W. Metric learning for multi-output tasks. *TPAMI*, 41(2):408–422, 2019.

Liu, Y., Jiang, S., and Liao, S. Efficient approximation of cross-validation for kernel methods using bouligand influence function. In *ICML*, pp. 324–332, 2014.

Liu, Z., Jin, C., Zhang, Y., and Zhang, T. Automatic naming of speakers in video via name-face mapping. In *NLP-NABD*, pp. 424–436, 2016.

Lv, J., Xu, M., Feng, L., Niu, G., Geng, X., and Sugiyama, M. Progressive identification of true labels for partial-label learning. In *ICML*, volume 119, pp. 6500–6510. PMLR, 2020.

Lyu, G., Feng, S., and Lang, C. A self-paced regularization framework for partial-label learning. *CoRR*, abs/1804.07759, 2018.

Lyu, G., Feng, S., Li, Y., Jin, Y., Dai, G., and Lang, C. HERA: partial label learning by combining heterogeneous loss with sparse and low-rank regularization. *ACM Transaction on Intelligent Systems and Technology*, 11 (3):34:1–34:19, 2020.

Lyu, G., Feng, S., Wang, T., Lang, C., and Li, Y. GM-PLL: graph matching based partial label learning. *IEEE Transactions on Knowledge and Data Engineering*, 33 (2):521–535, 2021.

Nguyen, N. and Caruana, R. Classification with partial labels. In *KDD*, pp. 551–559, 2008.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Köpf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pp. 8024–8035, 2019.

Pearlmutter, B. A. Fast exact multiplication by the hessian. *Neural Comput.*, 6(1):147–160, 1994.

Song, H., Wang, P., Yun, J., Li, W., Xu, B., and Wu, G. A weighted topic model learned from local semantic space for automatic image annotation. *IEEE Access*, 8:76411–76422, 2020.

Su, X., Zhou, H., Draghici, V. P., and Rätsch, M. Face naming in news images via multiple instance learning and hybrid recurrent convolutional neural network. *Journal of Electronic Imaging*, 27(03):033–036, 2018.

Wang, D., Li, L., and Zhang, M. Adaptive graph guided disambiguation for partial label learning. In *KDD*, pp. 83–91, 2019.

Xu, N., Lv, J., and Geng, X. Partial label learning via label enhancement. In *AAAI*, pp. 5557–5564, 2019.

Yan, Y. and Guo, Y. Partial label learning with batch label correction. In *AAAI*, pp. 6575–6582, 2020.

Yao, Y., Deng, J., Chen, X., Gong, C., Wu, J., and Yang, J. Deep discriminative CNN with temporal ensembling for ambiguously-labeled image classification. In *AAAI*, pp. 12669–12676, 2020.

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

Zeng, Z., Xiao, S., Jia, K., Chan, T., Gao, S., Xu, D., and Ma, Y. Learning by associating ambiguously labeled images. In *CVPR*, pp. 708–715, 2013.

Zhang, M. and Yu, F. Solving the partial label learning problem: An instance-based approach. In *IJCAI*, pp. 4048–4054, 2015.

Zhang, M., Zhou, B., and Liu, X. Partial label learning via feature-aware disambiguation. In *KDD*, pp. 1335–1344, 2016.

Zhou, Y., He, J., and Gu, H. Partial label learning via gaussian processes. *IEEE Transactions on Cybernetics*, 47(12):4443–4450, 2017.

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

A. Derivation of Eq. (9)

Let $h(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\ell(\mathbf{y}_i, f(\mathbf{x}_i;\mathbf{w}))$, $\hat{h}(\mathbf{w}) = h(\mathbf{w}) + \epsilon\ell(\mathbf{y}, f(\mathbf{x},\mathbf{w}))$, we then have

$$\hat{\mathbf{w}}_{\epsilon,-\mathbf{z}} = \arg\min_{\mathbf{w}} \hat{h}(\mathbf{w}) \tag{15}$$

Since $\hat{\mathbf{w}}_{\varepsilon,-\mathbf{z}}$ is a minimizer of Eq. (15), we can get the first order optimality condition as follows: $0 = \nabla h(\hat{\mathbf{w}}_{\epsilon,-\mathbf{z}}) + \epsilon\nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x};\hat{\mathbf{w}}_{\varepsilon,-\mathbf{z}}))$.

Since $\hat{\mathbf{w}}_{\epsilon,-\mathbf{z}} \to \hat{\mathbf{w}}$ as $\epsilon \to 0$, we perform a Taylor expansion of the right-hand side as follows: $0 \approx [\nabla h(\hat{\mathbf{w}}) + \epsilon\nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x};\hat{\mathbf{w}})] + \left[\nabla^2 h(\hat{\mathbf{w}}) + \epsilon\nabla^2_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x};\hat{\mathbf{w}})\right]\Delta\hat{\mathbf{w}}_j$.

Solving for $\Delta\hat{\mathbf{w}}_j$, we can get

$$\begin{aligned}\Delta\hat{\mathbf{w}}_j \approx &-\left[\nabla^2 h(\hat{\mathbf{w}}) + \epsilon\nabla^2_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x};\hat{\mathbf{w}})\right]^{-1} \\ &[\nabla h(\hat{\mathbf{w}}) + \epsilon\nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x};\hat{\mathbf{w}})]\end{aligned} \tag{16}$$

Since $\hat{\mathbf{w}}$ minimizes $h(\hat{\mathbf{w}})$, we have $\nabla h(\hat{\mathbf{w}}) = 0$. Removing $o(\epsilon)$ terms, we can get

$$\begin{aligned}&\Delta\hat{\mathbf{w}}_j \\ &\approx -\nabla^{-2}h(\hat{\mathbf{w}})\nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x};\hat{\mathbf{w}})\epsilon \\ &= -H^{-1}_{\hat{\mathbf{w}}}\nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x};\hat{\mathbf{w}}))\epsilon\end{aligned} \tag{17}$$

where $H_{\hat{\mathbf{w}}} = \nabla^2 h(\hat{\mathbf{w}}) = \frac{1}{n}\sum_{i=1}^{n}\nabla^2_{\mathbf{w}}\ell(\mathbf{y}_i, f(\mathbf{x}_i;\hat{\mathbf{w}}))$ is the Hessian matrix of $h(\hat{\mathbf{w}})$ and is positive definite by assumption. Subsequently, we can get

$$\frac{d\Delta\hat{\mathbf{w}}_j}{d\epsilon} = -H^{-1}_{\hat{\mathbf{w}}}\nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x};\hat{\mathbf{w}})) \tag{18}$$

We then define the label influence function of upweighting $\mathbf{z}$ on the predictive model optimizer $\hat{\mathbf{w}}$ as

$$\mathcal{I}(\mathbf{z}) = \frac{d\Delta\hat{\mathbf{w}}_j}{d\epsilon} = \frac{d\hat{\mathbf{w}}_{\epsilon,-\mathbf{z}}}{d\epsilon} = -H^{-1}_{\hat{\mathbf{w}}}\nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x};\hat{\mathbf{w}}))$$

According to Eq. (17) and the definition of label influence function, we can get $\Delta\hat{\mathbf{w}}_j = \epsilon\mathcal{I}(\mathbf{z})$.

B. Derivation of Eq. (11)

Let $g(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}\ell(\mathbf{y}_i, f(\mathbf{x}_i;\mathbf{w}))$, $\hat{g}(\mathbf{w}) = g(\mathbf{w}) - \varepsilon\ell(\mathbf{y}, f(\mathbf{x},\mathbf{w})) + \varepsilon\ell(\mathbf{y}_\delta, f(\mathbf{x},\mathbf{w}))$, we then have

$$\hat{\mathbf{w}}_{\varepsilon,-\mathbf{z},+\mathbf{z}_\delta} = \arg\min_{\mathbf{w}} \hat{g}(\mathbf{w}) \tag{19}$$

Since $\hat{\mathbf{w}}_{\varepsilon,-\mathbf{z},+\mathbf{z}_\delta}$ is a minimizer of Eq. (19), we can get the first order optimality condition as follows: $0 = \nabla g(\hat{\mathbf{w}}_{\varepsilon,-\mathbf{z},+\mathbf{z}_\delta}) - \varepsilon\nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x};\hat{\mathbf{w}}_{\varepsilon,-\mathbf{z},+\mathbf{z}_\delta})) + \varepsilon\nabla_{\mathbf{w}}\ell(\mathbf{y}_\delta, f(\mathbf{x};\hat{\mathbf{w}}_{\varepsilon,-\mathbf{z},+\mathbf{z}_\delta}))$.

Since $\hat{\mathbf{w}}_{\varepsilon,-\mathbf{z},+\mathbf{z}_\delta} \to \hat{\mathbf{w}}$ as $\varepsilon \to 0$, we perform a Taylor expansion of the right-hand side as follows: $0 \approx [\nabla g(\hat{\mathbf{w}}) - \varepsilon\nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x};\hat{\mathbf{w}}) + \varepsilon\nabla_{\mathbf{w}}\ell(\mathbf{y}_\delta, f(\mathbf{x};\hat{\mathbf{w}})] + \left[\nabla^2 g(\hat{\mathbf{w}}) - \varepsilon\nabla^2_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x};\hat{\mathbf{w}}) + \varepsilon\nabla^2_{\mathbf{w}}\ell(\mathbf{y}_\delta, f(\mathbf{x};\hat{\mathbf{w}})\right]\Delta\hat{\mathbf{w}}_l$.

Solving for $\Delta\hat{\mathbf{w}}_l$, we can get

$$\begin{aligned}&\Delta\hat{\mathbf{w}}_l \\ &\approx -\left[\nabla^2 g(\hat{\mathbf{w}}) - \varepsilon\nabla^2_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x};\hat{\mathbf{w}}) + \varepsilon\nabla^2_{\mathbf{w}}\ell(\mathbf{y}_\delta, f(\mathbf{x};\hat{\mathbf{w}})\right]^{-1} \\ &[\nabla g(\hat{\mathbf{w}}) - \varepsilon\nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x};\hat{\mathbf{w}}) + \varepsilon\nabla_{\mathbf{w}}\ell(\mathbf{y}_\delta, f(\mathbf{x};\hat{\mathbf{w}})]\end{aligned} \tag{20}$$

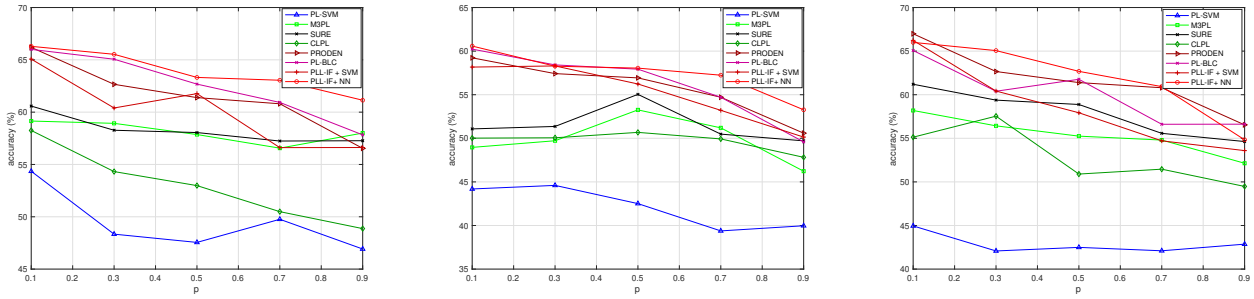
Since $\hat{\mathbf{w}}$ minimizes $g(\hat{\mathbf{w}})$, we have $\nabla g(\hat{\mathbf{w}}) = 0$. Removing $o(\varepsilon)$ terms, we can get

$$\begin{aligned}&\Delta\hat{\mathbf{w}}_l \\ &\approx -\nabla^{-2}g(\hat{\mathbf{w}})\left[\nabla_{\mathbf{w}}\ell(\mathbf{y}_\delta, f(\mathbf{x};\hat{\mathbf{w}}) - \nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x};\hat{\mathbf{w}})\right]\varepsilon \\ &= -H^{-1}_{\hat{\mathbf{w}}}\big(\nabla_{\mathbf{w}}\ell(\mathbf{y}_\delta, f(\mathbf{x};\hat{\mathbf{w}})) - \nabla_{\mathbf{w}}\ell(\mathbf{y}, f(\mathbf{x};\hat{\mathbf{w}}))\big)\varepsilon\end{aligned} \tag{21}$$

where $H_{\hat{\mathbf{w}}} = \nabla^2 g(\hat{\mathbf{w}}) = \frac{1}{n}\sum_{i=1}^{n} \nabla^2_{\mathbf{w}} \ell(\mathbf{y}_i, f(\mathbf{x}_i; \hat{\mathbf{w}}))$ is the Hessian matrix of $g(\hat{\mathbf{w}})$ and is positive definite by assumption.

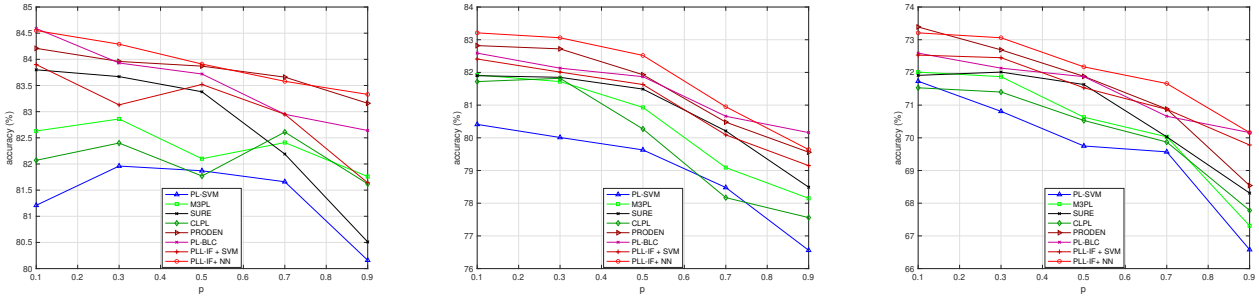
Similar to the definition of label influence function in the derivation of Eq. (9), we can get the influence of upweighting $\mathbf{z}_\delta$ and $\mathbf{z}$ on the predictive model optimizer $\hat{\mathbf{w}}$ as $\mathcal{I}(\mathbf{z}_\delta) = -H^{-1}_{\hat{\mathbf{w}}} \nabla_{\mathbf{w}} \ell(\mathbf{y}_\delta, f(\mathbf{x}; \hat{\mathbf{w}}))$ and $\mathcal{I}(\mathbf{z}) = -H^{-1}_{\hat{\mathbf{w}}} \nabla_{\mathbf{w}} \ell(\mathbf{y}, f(\mathbf{x}; \hat{\mathbf{w}}))$ respectively. And we then can approximate the l-th label impact (any of the other candidate labels) by $\Delta \hat{\mathbf{w}}_l = \varepsilon(\mathcal{I}(\mathbf{z}_\delta) - \mathcal{I}(\mathbf{z}))$.

C. Figs. 2 to 7



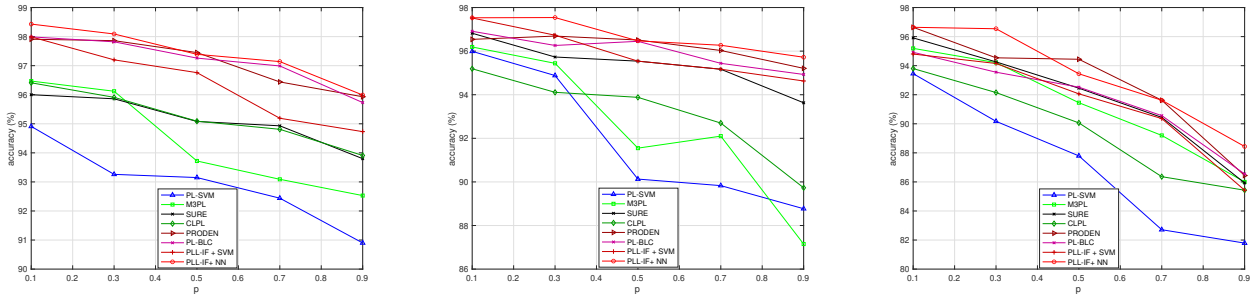
(a) r=1 (b) r=2 (c) r=3

Figure 2: Performance evaluation of the proposed PLL-IF and baselines with respect to the increasing proportion (p) of partially labelled examples in terms of mean accuracy (%) on *Glass* dataset.



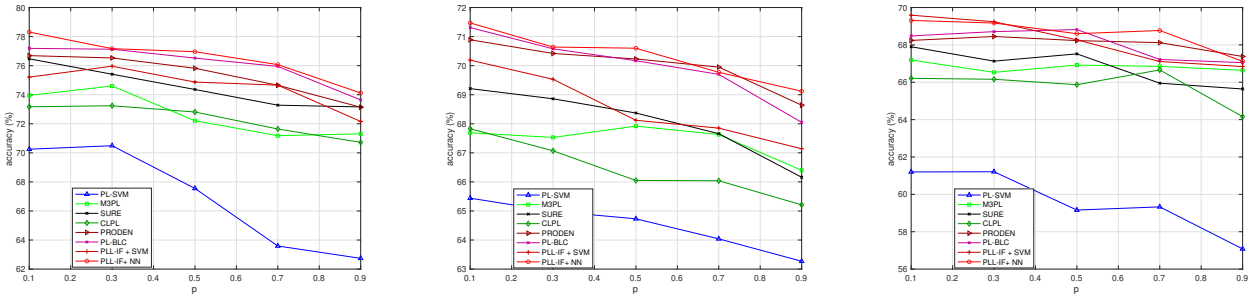
(a) r=1 (b) r=2 (c) r=3

Figure 3: Performance evaluation of the proposed PLL-IF and baselines with respect to the increasing proportion (p) of partially labelled examples in terms of mean accuracy (%) on *Ecoli* dataset.



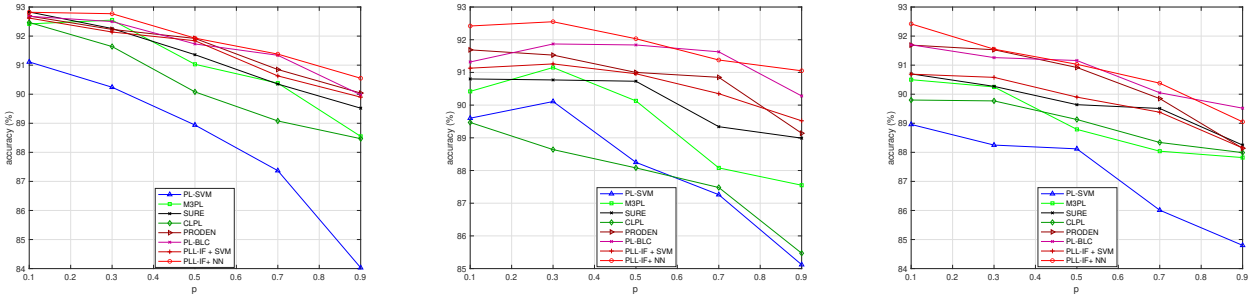
(a) r=1 (b) r=2 (c) r=3

Figure 4: Performance evaluation of the proposed PLL-IF and baselines with respect to the increasing proportion (p) of partially labelled examples in terms of mean accuracy (%) on *Dermatology* dataset.



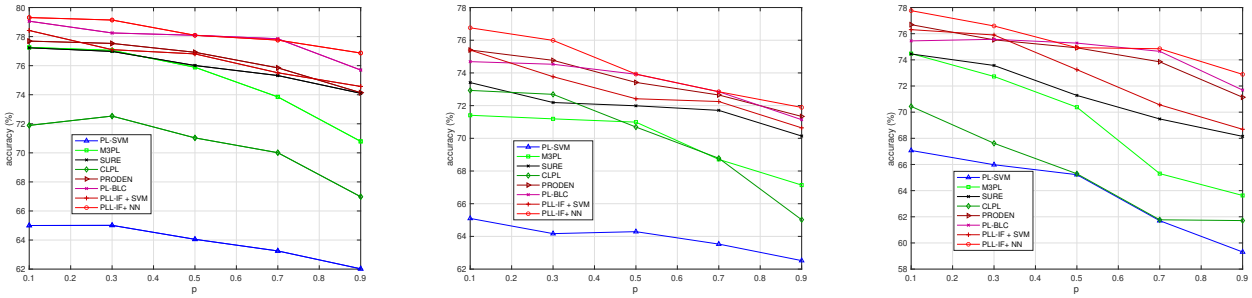
(a) r=1 (b) r=2 (c) r=3

Figure 5: Performance evaluation of the proposed PLL-IF and baselines with respect to the increasing proportion (p) of partially labelled examples in terms of mean accuracy (%) on *Vehicle* dataset.



(a) r=1 (b) r=2 (c) r=3

Figure 6: Performance evaluation of the proposed PLL-IF and baselines with respect to the increasing proportion (p) of partially labelled examples in terms of mean accuracy (%) on *Segmentation* dataset.



(a) r=1 (b) r=2 (c) r=3

Figure 7: Performance evaluation of the proposed PLL-IF and baselines with respect to the increasing proportion (p) of partially labelled examples in terms of mean accuracy (%) on *Satellite* dataset.