# Leveraged Weighted Loss for Partial Label Learning

**Hongwei Wen** [2 *] **Jingyi Cui** [1 *] **Hanyuan Hang** [2] **Jiabin Liu** [3] **Yisen Wang** [1] **Zhouchen Lin** [1 4]

## Abstract

As an important branch of weakly supervised learning, partial label learning deals with data where each instance is assigned with a set of candidate labels, whereas only one of them is true. Despite many methodology studies on learning from partial labels, there still lacks theoretical understandings of their risk consistent properties under relatively weak assumptions, especially on the link between theoretical results and the empirical choice of parameters. In this paper, we propose a family of loss functions named *Leveraged Weighted* (LW) loss, which for the first time introduces the leverage parameter $\beta$ to consider the trade-off between losses on partial labels and non-partial ones. From the theoretical side, we derive a generalized result of risk consistency for the LW loss in learning from partial labels, based on which we provide guidance to the choice of the leverage parameter $\beta$. In experiments, we verify the theoretical guidance, and show the high effectiveness of our proposed LW loss on both benchmark and real datasets compared with other state-of-the-art partial label learning algorithms.

## 1. Introduction

Partial label learning (Cour et al., 2011), also called ambiguously label learning (Chen et al., 2017) and superset label problem (Gong et al., 2017), refers to the task where each training example is associated with a set of candidate labels, while only one is assumed to be true. It naturally arises in a number of real-world scenarios such as web mining (Luo & Orabona, 2010), multimedia contents analysis (Cour et al., 2009; Zeng et al., 2013), ecoinformatics (Liu & Dietterich, 2012), etc, and subsequently attracts a lot of attention on methodology studies (Feng et al., 2020b; Wang & Zhang, 2020; Yao et al., 2020; Lyu et al., 2019; Wang et al., 2019).

As the main target of partial learning lies in disambiguating the candidate labels, two general strategies have been proposed with different assumptions to the latent label space: *1)* Average-based strategy that treats each candidate label equally in the model training phase (Hüllermeier & Beringer, 2006; Cour et al., 2011; Zhang & Yu, 2015). *2)* Identification-based strategy that considers the ground-truth label as a latent variable, and assume certain parametric model to describe the scores of each candidate label (Feng & An, 2019; Yan & Guo, 2020; Yao et al., 2020). The former is intuitive but has an obvious drawback that the predictions can be severely distracted by the false positive labels. The latter one attracted lots of attentions in the past decades but is criticized for the vulnerability when encountering differentiated label in candidate label sets. Furthermore, in recent years, more and more literature focuses on making amendments and adjustments on the optimization terms and loss functions on the basis of identification-based model (Lv et al., 2020; Cabannes et al., 2020; Wu & Zhang, 2018; Lyu et al., 2019; Feng et al., 2020b).

Despite extensive studies on partial label learning algorithms, theoretically guaranteed ones remain to be the minority. Some researchers have studied the statistical consistency (Cour et al., 2011; Feng et al., 2020b; Cabannes et al., 2020) and the learnability (Liu & Dietterich, 2014) of partial label learning algorithms. However, these theoretical studies are often based on rather strict assumptions, e.g. convexity of loss function (Cour et al., 2011), uniformly sampled partial label sets (Feng et al., 2020b), etc. Moreover, it remains to be an open problem why an algorithm performs better than others under specific parameter settings, or in other words, how can theoretical results guide parameter selections in computational implementations.

In this paper, we aim at investigating further theoretical explanations for partial label learning algorithms. Applying the basic structure of identification-based methods, we propose a family of loss functions named *Leveraged Weighted* (LW) loss. From the perspective of risk consistency, we provide theoretical guidance to the choice of the leverage parameter in our proposed LW loss by discussing the super-

[*]Equal contribution [1]Key Lab. of Machine Perception (MoE), School of EECS, Peking University, China [2]Department of Applied Mathematics, University of Twente, The Netherlands [3]Samsung Research China-Beijing, Beijing, China [4]Pazhou Lab, Guangzhou, China. Correspondence to: Yisen Wang <yisen.wang@pku.edu.cn>, Zhouchen Lin <zlin@pku.edu.cn>, Jiabin Liu <Jiabin.liu@samsung.com>.

*Proceedings of the $38^{th}$ International Conference on Machine Learning*, PMLR 139, 2021. 

vised loss to which LW is risk consistent. Then we design the partial label learning algorithm by iteratively identifying the weighting parameters. As follows are our contributions:

- We propose a family of loss function for partial label learning, named the Leveraged Weighted (LW) loss function, where we for the first time introduce the leverage parameter $\beta$ that considers the trade-offs between losses on partial labels and non-partial labels.
- We for the first time generalize the uniform assumption on the generation procedure of partial label sets, under which we prove the risk consistency of the LW loss. We also prove the Bayes consistency of our LW loss. Through discussions on the supervised loss to which LW is risk consistent, we obtain the potentially effective values of $\beta$.
- We present empirical understandings to verify the theoretical guidance to the choice of $\beta$, and experimentally demonstrate the effectiveness of our proposed algorithm based on the LW loss over other state-of-the-art partial label learning methods on both benchmark and real datasets.

## 2. Related Works

We briefly review the literature for partial label learning.

**Average-based methods.** The average-based methods normally consider each candidate label as equally important during model training, and average the outputs of all the candidate labels for predictions. Some researchers apply nearest neighbor estimators and predict a new instance by voting (Hüllermeier & Beringer, 2006; Zhang & Yu, 2015). Others further take advantage of the information in non-candidate samples. For example, (Cour et al., 2011; Zhang et al., 2016) employ parametric models to demonstrate the functional relationship between features and the ground truth label. The parameters are trained to maximize the average scores of candidate labels minus the average scores of non-candidate labels.

**Identification-based methods.** The identification-based methods aim at directly maximizing the output of exactly one candidate label, chosen as the truth label. A wealth of literature adopt major machine learning techniques such as maximum likelihood criterion (Jin & Ghahramani, 2002; Liu & Dietterich, 2012) and maximum margin criterion (Nguyen & Caruana, 2008; Yu & Zhang, 2016). As deep neural networks (DNNs) become popular, DNN-based methods outburst recently. (Feng & An, 2019) introduces self-learning with network structure; (Yan & Guo, 2020) studies the utilization of batch label correction; (Yao et al., 2020) manages to improve the performance by combining different networks. Moreover, it is worth highlighting that these algorithms have shown their weaknesses when facing the false positive labels that co-occur with the ground truth label.

**Binary loss-based multi-class classification.** Building multi-class classification loss from multiple binary ones is a general and frequently used scheme. In previous works, to extend margin-based binary classifiers (e.g., SVM and AdaBoost) to the multi-class setting, they adopted the combination of binary classification losses using constraint comparison (Lee et al., 2004; Zhang, 2004), loss-based decoding (Allwein et al., 2000), etc. In this paper, inspired by these losses for multi-class classification, we design a loss function for multi-class partial label learning via multiple binary loss functions.

In this paper, we follow the idea of the identification-based method, propose the LW loss function, and provide theoretical results on risk consistency. This result gives theoretical insights into the problem why an algorithm shows better performance under certain parameter settings than others.

## 3. Methodology

In this section, we first introduce some background knowledge about learning with partial labels in Section 3.1. Then in Section 3.2 we propose a family of LW loss function for partial labels. In Section 3.3, we prove the risk consistency of the LW loss and present guidance to the empirical choice of the leverage parameter $\beta$. Finally, we present our proposed practical algorithm in Section 3.4.

### 3.1. Preliminaries

**Notations.** Denote $\mathcal{X} \subset \mathbb{R}^d$ as a non-empty feature space (input space), $\mathcal{Y} = [K] := \{1, \dots, K\}$ as the *supervised label space*, where $k$ is the number of classes, and $\vec{\mathcal{Y}} := \{\vec{y} \,|\, \vec{y} \subset \mathcal{Y}\} = 2^{[K]}$ as the *partial label space*, where $2^{[K]}$ is the collection of all subsets in $[K]$. For the rest of this paper, $y$ denotes the true label of $x$ unless otherwise specified.

**Basic settings.** In learning with partial labels, an input variable $X \in \mathcal{X}$ is associated with a set of potential labels $\vec{Y} \in \vec{\mathcal{Y}}$ instead of a unique true label $Y \in \mathcal{Y}$. The goal is to find the latent *ground-truth label* $Y$ for the input $X$ through observing the *partial label set* $\vec{Y}$. The basic definition for partially supervised learning lies in the fact that the true label $Y$ of an instance $X$ must always reside in the partial label set $\vec{Y}$, i.e.

$$\mathrm{P}(y \in \vec{Y} \,|\, Y = y, x) = 1. \tag{1}$$

That is, we have $\#|\vec{Y}| \geq 1$, and $\#|\vec{Y}| = 1$ holds if and only if $\vec{Y} = \{y\}$, in which case the partial label learning problem reduces to multi-class classification with supervised labels.

**Risk consistency.** *Risk consistency* is an important tool in studying weakly supervised algorithms (Ishida et al.,

2017; 2019; Feng et al., 2020a;b). We say a method is *risk-consistent* if its corresponding classification *risk*, also called *generalization error*, is equivalent to the supervised classification risk $\mathcal{R}(f)$ given the same classifier $f$. Note that risk consistency implies *classifier consistency* (Xia et al., 2019), where learning from partial labels results in the same optimal classifier as that when learning from the fully supervised data.

To be specific, denote $g(x) = (g_1(x), \ldots, g_K(x))$ as the score function learned by an algorithm, where $g_z(x)$ is the score function for label $z \in [K]$. Larger $g_z(x)$ implies that $x$ is more likely to come from class $z \in [K]$. Then the resulting classifier is $f(x) = \arg\max_{z\in[K]} g_z(x)$. By definition, we denote

$$\mathcal{R}(\mathcal{L}, g) := \mathbb{E}_{(X,Y)}[\mathcal{L}(Y, g(X))], \tag{2}$$

as the *supervised risk* w.r.t. *supervised loss function* $\mathcal{L} : \mathcal{Y} \times \mathbb{R}^K \to \mathbb{R}^+$ for supervised classification learning. On the other hand, we denote

$$\bar{\mathcal{R}}(\bar{\mathcal{L}}, g) := \mathbb{E}_{(X,\vec{Y})}[\bar{\mathcal{L}}(\vec{Y}, g)] \tag{3}$$

as the *partial risk* w.r.t. *partial loss function* $\bar{\mathcal{L}} : \vec{\mathcal{Y}} \times \mathbb{R}^K \to \mathbb{R}^+$, measuring the expected loss of $g$ learned through partial labels w.r.t. the joint distribution of $(X, \vec{Y})$. Then a partial loss $\bar{\mathcal{L}}$ is *risk-consistent* to the supervised loss $\mathcal{L}$ if $\bar{\mathcal{R}}(\bar{\mathcal{L}}, g) = \mathcal{R}(\mathcal{L}, g)$.

**Bayes consistency.** We denote $g_L^* := \sup_{g\in\mathcal{M}} \mathcal{R}(\mathcal{L}, g)$ as the Bayes decision function w.r.t. the loss function $\mathcal{L}$, where $\mathcal{M}$ contains all measurable functions and $\mathcal{R}_{\mathcal{L}}^* := \mathcal{R}(\mathcal{L}, g^*)$. Similarly, we denote $\mathcal{R}^* := \mathcal{R}_{\mathcal{L}_{0\text{-}1}}^*$ as the Bayes decision function w.r.t. the multi-class 0-1 loss, i.e.

$$\mathcal{L}_{0\text{-}1}(y, g(x)) := \mathbf{1}\{\arg\max_{k\in[K]} g_k(x) \neq y\}, \tag{4}$$

where $\mathbf{1}\{\cdot\}$ denotes the indicator function. Then if there exist a collection $\{g_n\}$ such that $\mathcal{R}(\mathcal{L}_{0\text{-}1}, g_n) \to \mathcal{R}^*$ as $n \to \infty$, we say that the surrogate loss $\mathcal{L}$ reaches *Bayes risk consistency*.

### 3.2. Leveraged Weighted (LW) Loss Function

In this paper, we propose a family of loss function for partial label learning named *Leveraged Weighted* (LW) loss function. We adopt a multiclass scheme frequently used for the fully supervised setting (Crammer & Singer, 2001; Rifkin & Klautau, 2004; Zhang, 2004; Tewari & Bartlett, 2005), which combines binary losses $\psi(\cdot) : \mathbb{R} \to \mathbb{R}^+$, a non-increasing function, to create a multiclass loss. We highlight that it is the first time that the leverage parameter $\beta$ is introduced into loss functions for partial label learning, which leverages between losses on partial labels and non-partial ones. To be specific, the partial loss function of concern is of the form

$$\bar{\mathcal{L}}_\psi(\vec{y}, g(x)) = \sum_{z\in\vec{y}} w_z\psi(g_z(x)) + \beta \cdot \sum_{z\notin\vec{y}} w_z\psi(-g_z(x)), \tag{5}$$

where $\vec{y} \in \vec{\mathcal{Y}}$ denotes the partial label set. It consists of three components.

- A binary loss function $\psi(\cdot) : \mathbb{R} \to \mathbb{R}^+$, where $\psi(g_z(x))$ forces $g_z$ to be larger when $z$ resides in the partial label set $\vec{y}$, while $\psi(-g_z(x))$ punishes large $g_z$ when $z \notin \vec{y}$.
- Weighting parameters $w_z \geq 0$ on $\psi(g_z)$ for $z \in [K]$. Generally speaking, we would like to assign more weights to the loss of labels that are more likely to be the true label.
- The leverage parameter $\beta \geq 0$ that distinguishes between partial labels and non-partial ones. Larger $\beta$ quickly rules out non-partial labels during training, while it also lessens weights assigned to partial labels.

We mention that the partial loss proposed in (5) is a general form. Some special cases include

*1)* Taking $\beta = 0$, $w_z = 1/\#|\vec{y}|$ for $z \in \vec{y}$, we achieve the partial loss proposed by (Jin & Ghahramani, 2002), the form of which is

$$\frac{1}{\#|\vec{y}|}\sum_{y\in\vec{y}} \psi(g_y(x)). \tag{6}$$

*2)* Taking $\beta = 0$, and $w_{z^*} = 1$ where $z^* = \arg\max_{z\in\vec{y}} g_z$, $w_z = 0$ for $z \in \vec{y}\backslash\{z^*\}$, we achieve the partial loss function proposed by (Lv et al., 2020), with the form

$$\psi(\max_{y\in\vec{y}} g_y(x)) \Leftrightarrow \min_{y\in\vec{y}} \psi(g_y(x)). \tag{7}$$

*3)* By taking $\beta = 1$, and $w_{z^*} = 1$ where $z^* = \arg\max_{z\in\vec{y}} g_z$, $w_z = 0$ for $z \in \vec{y} \setminus \{z^*\}$, $w_z = 1$ for $z \notin \vec{y}$, we achieve the partial loss function proposed by (Cour et al., 2011), with the form

$$\psi(\max_{y\in\vec{y}} g_y(x)) + \sum_{y\notin\vec{y}} \psi(-g_y(x)). \tag{8}$$

### 3.3. Theoretical Interpretations

In this part, we first relax the assumption on the generation procedure of the partial label set, and show the risk consistency of our proposed LW loss function. Then by observing the supervised loss to which LW is risk consistent, we study the leverage parameter $\beta$ and deduce its reasonable values. All proofs are shown in Section A of the supplements.

#### 3.3.1. Generalizing the Uniform Sampling Assumption

In previous study of risk consistency, the partial label set $\vec{Y}$ is assumed to be independently and uniformly sampled given a specific true label $Y$ (Feng et al., 2020b), i.e.

$$\mathrm{P}(\vec{Y}=\vec{y}\,|\,Y=y,x)=\begin{cases}\dfrac{1}{2^{k-1}-1}, & \text{if } y\in\vec{y},\\ 0, & \text{otherwise.}\end{cases} \tag{9}$$

Note that this data generation procedure is equivalent to assuming $\mathrm{P}(y\in\vec{Z}\,|\,x)=\frac{1}{2}$, where $\vec{Z}$ is an unknown label set uniformly sampled from $[K]$. The intuition behind is that if no information of $\vec{Z}$ is given, we may randomly guess with even probabilities whether the correct $y$ is included in an unknown label set $\vec{Z}$ or not.

However, in real-world situations, some combination of partial labels may be more likely to appear than others. Instances belonging to certain classes usually share similar features e.g. images of *dog* and *cat* may look alike, while they may be less similar to images of *truck*. Thus, given these shared features indicating the true label of an instance, the probability of label $z\neq y$ entering the partial label set may be different. For instance, when the true label is *dog*, *cat* is more likely to be picked as a partial label than *truck*.

Therefore, in this paper, we generalize the uniform sampling of partial label sets, and allow the sampling probability to be *label-specific*. Denote $q_z\in[0,1]$ as

$$q_z:=\mathrm{P}(z\in\vec{Y}\,|\,Y=y,x), \tag{10}$$

for $z\in[K]$. Then for $z=y$, we have $q_y=1$ according to the problem settings of learning from partial labels, and for $z\neq y$, we have $q_z<1$ due to the small ambiguity degree condition (Cour et al., 2011), which guarantees the ERM learnability of partial label learning problems (Liu & Dietterich, 2014; Lv et al., 2020). Then when the elements in $\vec{y}$ is assumed to be independently drawn, the conditional distribution of the partial label set $\vec{Y}$ turns out to be

$$\mathrm{P}(\vec{Y}=\vec{y}\,|\,Y=y,x)=\prod_{s\in\vec{y},s\neq y}q_s\cdot\prod_{t\notin\vec{y}}(1-q_t). \tag{11}$$

where $y$ is the true label of input $x$.

Note that the above generation procedure of the partial label set allows the existence of $[K]$ to be a partial label set. If we want to rule out this set, we can simply drop it and sample the partial label set again. By this means, the conditional distribution becomes

$$\mathrm{P}(\vec{Y}=\vec{y}\,|\,Y=y,x)=\frac{1}{1-M}\prod_{s\in\vec{y},s\neq y}q_s\cdot\prod_{t\notin\vec{y}}(1-q_t),$$

where $M=\prod_{z\neq y}q_z$. Taking the special case where $q_z=1/2$ for all $z\neq y$, we reduce to the generation procedure (9) as in (Feng et al., 2020b).

#### 3.3.2. Risk-consistent Loss Function

Under the above generation procedure, we take a deeper look at our proposed LW loss and prove its risk consistency.

**Theorem 1** *The LW partial loss function proposed in (5) is risk-consistent with respect to the supervised loss function with the form*

$$\mathcal{L}_\psi(y,g(x))=w_y\psi(g_y(x))+\sum_{z\neq y}w_zq_z\big[\psi(g_z(x))+\beta\psi(-g_z(x))\big]. \tag{12}$$

Theorem 1 indicates the existence of a loss function $\mathcal{L}_\psi$ for supervised learning to which the LW loss $\bar{\mathcal{L}}_\psi$ is risk consistent. Note that the resulting form of the supervised loss function (12) is a widely used multi-class scheme in supervised learning, e.g. Crammer & Singer (2001); Rifkin & Klautau (2004); Tewari & Bartlett (2007).

It is worth mentioning that this is the first time that a risk consistency analysis is conducted under a label-specific sampling of the partial label set. Moreover, compared with Lv et al. (2020), where the proposed loss function is proved to be classifier consistent under the deterministic scenario, our result on risk consistency is a stronger claim and applies to both deterministic and stochastic scenarios.

The next theorem shows that as long as $\beta>0$, the supervised risk induced by (12) is consistent to the Bayes risk $\mathcal{R}^*$. That is, optimizing the supervised loss in (12) can result in the Bayes classifier under 0-1 loss.

**Theorem 2** *Let $\mathcal{L}_\psi$ be of the form in (12) and $\mathcal{L}_{0\text{-}1}$ be the multi-class 0-1 loss. Assume that $\psi(\cdot)$ is differentiable and symmetric, i.e. $\psi(g_z(x))+\psi(-g_z(x))=1$. For $\beta>0$, if there exist a sequence of functions $\{\hat{g}_n\}$ such that*

$$\mathcal{R}(\mathcal{L}_\psi,\hat{g}_n)\to\mathcal{R}^*_{\mathcal{L}_\psi},$$

*then we have*

$$\mathcal{R}(\mathcal{L}_{0\text{-}1},\hat{g}_n)\to\mathcal{R}^*.$$

Combined with Theorem 1, when $\beta>0$, we have our LW loss consistent to the Bayes classifier.

#### 3.3.3. Guidance on the Choice of $\beta$

In this section, we try to answer the question why we should choose some certain values of $\beta$ for the LW loss $\bar{\mathcal{L}}_\psi$ instead of others when learning from partial labels. Recall that when minimizing a risk consistent partial loss function in partial label learning, we are at the same time minimizing the corresponding supervised loss. Therefore, by Theorem 1, a satisfactory supervised loss $\mathcal{L}_\psi$ in supervised learning

naturally corresponds to an LW loss $\bar{\mathcal{L}}_\psi$ with the desired value of the leverage parameter $\beta$ in partial label learning.

When we take a closer look at the right-hand side of (12), the loss function $\mathcal{L}_\psi$ to which LW loss is risk-consistent always contains the term $\psi(g_y)$, which focuses on identifying the true label $y$. On the other hand, an interesting finding is that the leverage parameter $\beta$ determines the relative scale of $\psi(g_z)$ and $\psi(-g_z)$ for all $z \neq y$, while it does not affect the loss on the true label $y$.

In the following discussions, we focus on symmetric binary loss $\psi(\cdot)$, where $\psi(g_z(x)) + \psi(-g_z(x)) = 1$, for their fine theoretical properties. We remark that commonly adopted loss functions such as zero-one loss, Sigmoid loss, Ramp loss, etc. satisfy the symmetric condition. In what follows, we present the results of risk consistency for LW loss with specific values of $\beta$, and discuss each case respectively.

**Case 1:** When $\beta = 0$ (e.g. Lv et al., 2020), the LW loss function $\bar{\mathcal{L}}_\psi$ is risk-consistent to

$$w_y \psi(g_y(x)) + \sum_{z \neq y} w_z q_z \psi(g_z(x)). \tag{13}$$

In this case, in addition to focusing on the true label $y$, $\mathcal{L}_\psi$ also gives positive weights to the untrue labels as long as there exists a label $z \neq y$ such that $w_z > 0$. Since the minimization of $\psi(g_z)$ may lead to false identification of label $z \neq y$, $\beta = 0$ is not preferred for LW loss.

**Case 2:** When $\beta = 1$ (e.g. Jin & Ghahramani, 2002; Cour et al., 2011), the LW loss function $\bar{\mathcal{L}}_\psi$ is risk-consistent to

$$w_y \psi(g_y(x)) + \sum_{z \neq y} w_z q_z. \tag{14}$$

In this case, the minimization of $\bar{\mathcal{L}}_\psi$ indicates the minimization of $\mathcal{L}_\psi = \psi(g_y(x))$, aiming at directly identifying the true label $y$. The idea is similar to that of the cross entropy loss, where $\mathcal{L}_{CE}(y, g(x)) := -\log(g_y(x))$ . Therefore, we take $\beta = 1$ as a reasonable choice for LW loss.

**Case 3:** When $\beta = 2$, the LW loss function $\bar{\mathcal{L}}_\psi$ is risk-consistent with

$$w_y \psi(g_y(x)) + \sum_{z \neq y} w_z q_z \psi(-g_z(x)) + \sum_{z \neq y} w_z q_z. \tag{15}$$

In this case, the LW loss not only encourages the learner to identify the true label $y$ by minimizing $\psi(g_y)$, but also helps rule out the untrue labels $z \neq y$ by punishing large value of $\psi(-g_z)$. Moreover, for a confusing label $z \neq y$ that is more likely to appear in the partial label set, i.e. $q_z$ is larger, $\mathcal{L}_\psi$ imposes severer punishment on $g_z$. Therefore, $\beta = 2$ is also a preferred choice for LW loss. Especially, when taking $w_z = 1/q_z$ for $z \in [K]$, we achieve the form

$$\psi(g_y(x)) + \sum_{z \neq y} \psi(-g_z(x)) + K - 1, \tag{16}$$

which exactly corresponds to the *one-versus-all* (OVA) loss function proposed by Zhang (2004).

To conclude, it is not a good choice for LW loss to take $\beta = 0$, as most commonly used loss functions do. Our theoretical interpretations of risk consistency show that $\beta > 0$ and especially $\beta > 1$ are preferred choices, which are also empirically verified in Section 4.2.1.

### 3.4. Practical Algorithm

In the theoretical analysis in the previous section, we focus on partial and supervised loss functions that are consistent in risk. However, in experiments, the risk for partial label loss is not directly accessible since the underlying distribution of $\mathrm{P}(X, \vec{Y})$ is unknown. Instead, on the partially labeled sample $D_n := \{(x_1, \vec{y}_1), \ldots, (x_n, \vec{y}_n)\}$, we try to minimize the empirical risk of a learning algorithm defined by

$$\bar{\mathcal{R}}_{D_n}(\bar{\mathcal{L}}, g(X)) = \frac{1}{n} \sum_{i=1}^{n} \bar{\mathcal{L}}(\vec{y}_i, g(x_i)). \tag{17}$$

Moreover, in this part we take the network parameters $\theta$ for score functions $g(x) := \big(g_1(x), \ldots, g_K(x)\big)$ into consideration, and write $g(x; \theta)$ and $g_z(x; \theta)$ instead.

**Determination of weighting parameters.** Since our goal is to find out the unique true label after observing partially labeled data, we'd like to focus more on the true label contained in the partial label set, while ruling out the most confusing one outside this set. Therefore, we assign larger weights to $\psi(g_y(x))$, where $y$ denotes the true label of $x$, and to $\psi(-g_z(x))$, where $z$ is the non-partial label with the highest score among $[K] \setminus \vec{y}$.

However, since we cannot directly observe the true label $y$ for input $x$ from the partially labeled data, the weighting parameters cannot be directly assigned. Therefore, inspired by the EM algorithm (Dempster et al., 1977) and PRODEN (Lv et al., 2020), we learn the weighting parameters through an iterative process instead of assigning fixed values.

To be specific, at the $t$-th step, given the network parameters $\theta^{(t)}$, we calculate the weighting parameters by respectively normalizing the score functions $g_z(x; \theta)$ for $z \in \vec{y}$ and those for $z \notin \vec{y}$, i.e.

$$w_z^{(t)} = \frac{\exp(g_z(x; \theta^{(t)}))}{\sum_{z \in \vec{y}} \exp(g_z(x; \theta^{(t)}))} \text{ for } z \in \vec{y}, \text{ and} \tag{18}$$

$$w_z^{(t)} = \frac{\exp(g_z(x; \theta^{(t)}))}{\sum_{z \notin \vec{y}} \exp(g_z(x; \theta^{(t)}))} \text{ for } z \notin \vec{y}. \tag{19}$$

By this means we have $\sum_{z \in \vec{y}} w_z^{(t)} = \sum_{z \notin \vec{y}} w_z^{(t)} = 1$. Note that $w_z^{(t)}$ varies with sample instances. Thus for each instance $(x_i, \vec{y}_i)$, $i = 1, \ldots, n$, we denote the weighting

parameter as $w_{z,i}^{(t)}$. As a special reminder, we initialize $w_{z,i}^{(0)} = \frac{1}{\#|\vec{y}_i|}$ for $z \in \vec{y}_i$ and $w_{z,i}^{(0)} = \frac{1}{K-\#|\vec{y}_i|}$ for $z \notin \vec{y}_i$.

The intuition behind the respective normalization is twofold. First of all, by respectively normalizing scores of partial labels and non-partial ones, we achieve our primary goal of focusing on the true label and the most confusing non-partial label. Secondly, if we simply perform normalization on all score functions, the weights for partial labels tend to grow rapidly through training, resulting in much larger weights for the partial losses than the non-partial ones. Thus, as the training epochs grow, the losses on non-partial labels as well as the leverage parameter $\beta$ gradually become ineffective, which we are not pleased to see.

The main algorithm is shown in Algorithm 1. Note that here $\beta$ is a hyper-parameter tuned by validation while $w$ is the parameter trained through data.

**Algorithm 1** LW Loss for Partial Label Learning

**Input:** Training data $D_n := \{(x_1, \vec{y}_1), \ldots, (x_n, \vec{y}_n)\}$;
Leverage parameter $\beta$;
Learning rate $\rho$;
Number of Training Epochs $T$;
**for** $t = 1$ **to** $T$ **do**
Calculate $\bar{\mathcal{R}}_{D_n}^{(t)}(\bar{\mathcal{L}}^{(t-1)}, g(x; \theta^{(t-1)}))$ by (17);
Update network parameter $\theta^{(t)}$ and achieve $g(x; \theta^{(t)})$.
Update weighting parameters $w_{z,i}^{(t)}$ by (18) and (19);
**end for**
**Output:** Decision function $\arg\max_{z \in [K]} g_z(x; \theta^{(T)})$.

## 4. Experiments

In this part, we empirically verify the effectiveness of our proposed algorithm through performance comparisons as well as other empirical understandings.

### 4.1. The Classification Performance

In this section, we conduct empirical comparisons with other state-of-the-art partial label learning algorithms on both benchmark and real datasets.

#### 4.1.1. Benchmark Dataset Comparisons

**Datasets.** We base our experiments on four benchmark datasets: MNIST (LeCun et al., 1998), Kuzushiji-MNIST (Clanuwat et al., 2018), Fashion-MNIST (Xiao et al., 2017), and CIFAR-10 (Krizhevsky et al., 2009). We generate partially labeled data by making $K - 1$ independent decisions for labels $z \neq y$, where each label $z$ has probability $q_z$ to enter the partial label set. In this part we consider $q_z = q$ for all $z \neq y$, where $q \in \{0.1, 0.3, 0.5\}$ and larger $q$ indicates that the partially labeled data is more ambiguous. We put the experiments based on non-uniform data generating procedures in Section 4.2.3. Note that the true label $y$ always resides in the partial label set $\vec{y}$ and we accept the occasion that $\vec{y} = [K]$. On MNIST, Kuzushiji-MNIST, and Fashion-MNIST, we employ the base model as a 5-layer perception (MLP). On the CIFAR-10 dataset, we employ a 12-layer ConvNet (Laine & Aila, 2016) for all compared methods. More details are shown in Section B.1 of the supplements.

**Compared methods.** We compare with the state-of-the-art PRODEN (Lv et al., 2020), RC and CC (Feng et al., 2020b), with all hyper-parameters searched according to the suggested parameter settings in the original papers. For our proposed method, we search the initial learning rate from $\{0.001, 0.005, 0.01, 0.05, 0.1\}$ and weight decay from $\{10^{-6}, 10^{-5}, \ldots, 10^{-2}\}$, with the exponential learning rate decay halved per 50 epochs. We search $\beta \in \{1, 2\}$ according to the theoretical guidance discussed in Section 3.3. For computational implementations, we use `PyTorch` (Paszke et al., 2019) and the stochastic gradient descent (SGD) (Robbins & Monro, 1951) optimizer with momentum 0.9. For all methods, we set the mini-batch size as 256 and train each model for 250 epochs. Hyper-parameters are searched to maximize the accuracy on a validation set containing 10% of the partially labeled training samples. We adopt the same base model for fair comparisons. More details are shown in Section B.2 of the supplements.

**Experimental results.** We repeat all experiments 5 times, and report the average accuracy and the standard deviation. We apply the Wilcoxon signed-rank test (Wilcoxon, 1992) at the significance level $\alpha = 0.05$. As is shown in Table 1, when adopting the Sigmoid loss function with fine symmetric theoretical property, our proposed LW loss outperforms almost all other state-of-the-art algorithms for learning with partial labels. Moreover, by adopting the widely used cross entropy loss function, the empirical performance of LW can be further significantly improved on MNIST, Fashion-MNIST, and Kuzushiji-MNIST datasets. We attribute this satisfactory result to the design of a proper leveraging parameter $\beta$, which makes it possible to consider the information of both partial labels and non-partial ones.

#### 4.1.2. Real Data Comparisons

**Datasets.** In this part we base our experimental comparisons on 5 real-world datasets including: Lost (Cour et al., 2011), MSRCv2 (Liu & Dietterich, 2012), BirdSong (Briggs et al., 2012), Soccer Player (Zeng et al., 2013), and Yahoo! News (Guillaumin et al., 2010).

**Compared methods.** Aside from the network-based methods mentioned in Section 4.1.1, we compare with 3 other state-of-the-art partial label learning algorithms including IPAL (Zhang & Yu, 2015), PALOC (Wu & Zhang, 2018), and PLECOC (Zhang et al., 2017), where the hyper-

*Table 1.* Accuracy comparisons on benchmark datasets.

| Dataset | Method | Base Model | $q = 0.1$ | $q = 0.3$ | $q = 0.5$ |
|---|---|---|---|---|---|
| MNIST | RC | MLP | $98.44 \pm 0.11\%*$ | $98.29 \pm 0.05\%*$ | $98.14 \pm 0.03\%*$ |
| | CC | MLP | $98.56 \pm 0.06\%*$ | $98.32 \pm 0.06\%*$ | $98.21 \pm 0.07\%*$ |
| | PRODEN | MLP | $98.57 \pm 0.07\%*$ | $98.48 \pm 0.10\%*$ | $98.40 \pm 0.15\%*$ |
| | LW-Sigmoid | MLP | $\underline{98.82 \pm 0.04\%}$ | $\underline{98.74 \pm 0.07\%}$ | $\underline{98.55 \pm 0.07\%}$ |
| | LW-Cross entropy | MLP | $\mathbf{98.89 \pm 0.06\%}$ | $\mathbf{98.81 \pm 0.06\%}$ | $\mathbf{98.59 \pm 0.15\%}$ |
| Fashion-MNIST | RC | MLP | $89.69 \pm 0.08\%*$ | $89.47 \pm 0.04\%*$ | $\underline{88.97 \pm 0.06\%}*$ |
| | CC | MLP | $89.63 \pm 0.10\%*$ | $89.11 \pm 0.19\%*$ | $88.31 \pm 0.14\%*$ |
| | PRODEN | MLP | $89.62 \pm 0.13\%*$ | $89.17 \pm 0.08\%*$ | $88.72 \pm 0.18\%*$ |
| | LW-Sigmoid | MLP | $\underline{90.25 \pm 0.16\%}$ | $\underline{89.67 \pm 0.15\%}*$ | $88.76 \pm 0.03\%*$ |
| | LW-Cross entropy | MLP | $\mathbf{90.52 \pm 0.19\%}$ | $\mathbf{90.15 \pm 0.13\%}$ | $\mathbf{89.54 \pm 0.10\%}$ |
| Kuzushiji-MNIST | RC | MLP | $92.12 \pm 0.17\%*$ | $91.83 \pm 0.18\%*$ | $90.84 \pm 0.26\%*$ |
| | CC | MLP | $92.57 \pm 0.14\%*$ | $92.08 \pm 0.06\%*$ | $90.58 \pm 0.18\%*$ |
| | PRODEN | MLP | $92.20 \pm 0.43\%*$ | $91.18 \pm 0.15\%*$ | $89.64 \pm 0.32\%*$ |
| | LW-Sigmoid | MLP | $\underline{93.63 \pm 0.39\%}$ | $\underline{92.92 \pm 0.28\%}*$ | $\underline{91.81 \pm 0.25\%}*$ |
| | LW-Cross entropy | MLP | $\mathbf{94.14 \pm 0.12\%}$ | $\mathbf{93.57 \pm 0.13\%}$ | $\mathbf{92.30 \pm 0.23\%}$ |
| CIFAR-10 | RC | ConvNet | $86.53 \pm 0.12\%*$ | $85.90 \pm 0.13\%*$ | $84.48 \pm 0.17\%*$ |
| | CC | ConvNet | $86.47 \pm 0.22\%*$ | $85.33 \pm 0.19\%*$ | $82.74 \pm 0.22\%*$ |
| | PRODEN | ConvNet | $89.71 \pm 0.13\%*$ | $88.57 \pm 0.10\%*$ | $85.95 \pm 0.14\%*$ |
| | LW-Sigmoid | ConvNet | $\mathbf{90.88 \pm 0.09\%}$ | $\mathbf{89.75 \pm 0.08\%}$ | $\underline{87.27 \pm 0.15\%}*$ |
| | LW-Cross entropy | ConvNet | $\underline{90.58 \pm 0.04\%}*$ | $\underline{89.68 \pm 0.10\%}$ | $\mathbf{88.31 \pm 0.09\%}$ |

The best results are marked in **bold** and the second best marked in underline. The standard deviation is also reported. We use $*$ to represent that the best method is significantly better than the other compared methods.

*Table 2.* Accuracy comparisons on real datasets.

| Method | Dataset | | | | |
|---|---|---|---|---|---|
| | Lost | MSRCv2 | Birdsong | SoccerPlayer | YahooNews |
| IPAL | $62.37 \pm 4.81\%*$ | $50.34 \pm 3.24\%*$ | $70.20 \pm 4.62\%*$ | $55.79 \pm 0.88\%*$ | $64.57 \pm 1.51\%*$ |
| PALOC | $57.80 \pm 7.00\%*$ | $47.51 \pm 3.78\%*$ | $70.20 \pm 3.79\%*$ | $53.96 \pm 2.38\%*$ | $60.36 \pm 1.48\%*$ |
| PLECOC | $63.04 \pm 6.72\%*$ | $44.13 \pm 5.06\%*$ | $73.88 \pm 3.41\%$ | $29.39 \pm 9.38\%*$ | $60.41 \pm 1.50\%*$ |
| RC-Linear | $75.93 \pm 3.62\%*$ | $45.82 \pm 4.74\%*$ | $71.73 \pm 2.84\%*$ | $57.00 \pm 2.15\%$ | $67.42 \pm 1.11\%*$ |
| CC-Linear | $75.57 \pm 3.58\%*$ | $45.56 \pm 3.97\%*$ | $71.83 \pm 2.85\%*$ | $56.75 \pm 1.87\%*$ | $67.43 \pm 1.07\%*$ |
| PRODEN-Linear | $76.33 \pm 4.51\%$ | $44.04 \pm 4.50\%*$ | $71.97 \pm 2.73\%*$ | $55.93 \pm 2.34\%*$ | $67.47 \pm 1.11\%*$ |
| LW-Linear | $\mathbf{76.50 \pm 4.16\%}$ | $\underline{46.34 \pm 2.72\%}*$ | $\underline{72.33 \pm 3.29\%}*$ | $\mathbf{57.29 \pm 2.37\%}$ | $\mathbf{68.67 \pm 1.05\%}$ |
| RC-MLP | $63.45 \pm 5.03\%*$ | $51.60 \pm 2.53\%*$ | $73.11 \pm 4.45\%*$ | $53.87 \pm 1.96\%*$ | $63.84 \pm 0.65\%*$ |
| CC-MLP | $65.29 \pm 4.15\%*$ | $50.97 \pm 3.05\%*$ | $70.97 \pm 3.66\%*$ | $\underline{54.03 \pm 1.77\%}*$ | $62.85 \pm 1.26\%*$ |
| PRODEN-MLP | $61.41 \pm 5.20\%*$ | $50.54 \pm 3.37\%*$ | $72.74 \pm 4.78\%*$ | $53.48 \pm 1.74\%*$ | $61.88 \pm 0.96\%*$ |
| LW-MLP | $\underline{66.00 \pm 4.10\%}*$ | $\mathbf{52.23 \pm 3.61\%}$ | $\mathbf{73.89 \pm 4.01\%}$ | $53.64 \pm 1.83\%*$ | $\underline{64.65 \pm 0.98\%}*$ |

The best results among all methods are marked in **bold** and the best under the same base model is marked in underline. The standard deviation is also reported. We use $*$ to represent that the best method is significantly better than the other compared methods.

parameters are searched through a 5-fold cross-validation under the suggested settings in the original papers. We adopt cross entropy loss for LW and employ both linear model and MLP as base models. For all compared methods, we adopt a 10-fold cross-validation to evaluate the testing performances. Other settings are similar to Section 4.1.1.

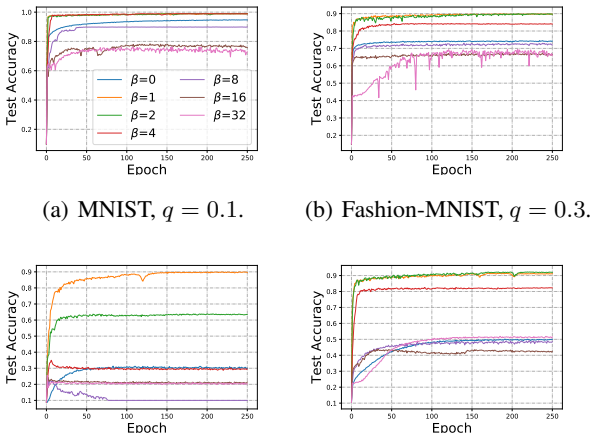
**Experimental results.** In Table 2, under the same base model, our proposed LW shows the best performance on almost all datasets. Moreover, on all real datasets, LW loss with proper base models always outperforms other state-of-the-art methods. Different from the benchmark datasets, the distribution of real partial labels remains unknown and could be more complex. Since our proposed LW loss is risk consistent with desired supervised loss functions under a generalized partial label generation assumption, there is no surprise that it presents satisfactory empirical performance.

### 4.2. Empirical Understandings

In this part, we conduct a series of comprehensive experiments to verify the effectiveness of our proposed LW loss.

#### 4.2.1. Parameter analysis

We study the leverage parameter $\beta$ of LW loss by comparing its performances under $\beta \in \{0, 1, 2, 4, 8, 16, 32\}$ respectively. We employ Sigmoid loss function for LW loss, and other experimental settings are similar to Section 4.1.1.




(a) MNIST, $q = 0.1$. (b) Fashion-MNIST, $q = 0.3$.

(c) CIFAR-10, $q = 0.3$. (d) Kuzushiji-MNIST, $q = 0.5$.

*Figure 1.* Study of the leverage parameter $\beta$ for LW loss.

As is shown in Figure 1, on all four datasets with varying data generation probability $q$, LW losses with $\beta = 1$ and $\beta = 2$ significantly outperform those with other parameter settings. (On MNIST, LW loss with $\beta = 4$ also performs competitively.) This coincides exactly with the theoretical guidance to the choice of $\beta$ discussed in Section 3.3.

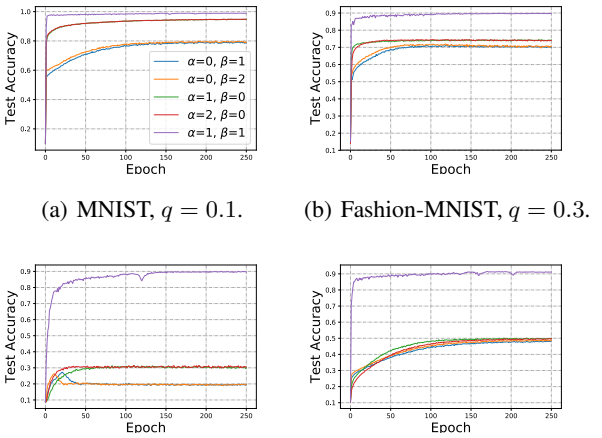
#### 4.2.2. Ablation Study

In this part, we conduct an ablation study on effect of the two parts in our proposed LW loss, i.e. losses on partial labels $\sum_{z\in\vec{y}} w_z\psi(g_z(x))$ and those on non-partial ones $\sum_{z\notin\vec{y}} w_z\psi(-g_z(x))$. For notational simplicity, we rewrite the "generalized" LW loss as

$$\alpha \cdot \sum_{z\in\vec{y}} w_z\psi(g_z(x)) + \beta \cdot \sum_{z\notin\vec{y}} w_z\psi(-g_z(x)).$$

We compare among performances of LW with
1) losses on partial labels only ($\beta = 0$),
2) losses on non-partial labels only ($\alpha = 0$),
3) losses on both partial and non-partial labels ($\alpha\beta \neq 0$).
We employ the Sigmoid loss function for the LW loss. Other experimental settings are similar to Section 4.1.1.

As is shown in Figure 2, when individually using losses on either partial labels or non-partial ones, the accuracy results is far from satisfactory on all three datasets since the information contained in the other half is neglected. Besides, it provides little help to the empirical performance by simply scaling the losses themselves. On the contrary, by combining losses on both partial labels and non-partial ones ($\alpha = \beta = 1$), our proposed LW loss function shows its superiority in empirical performances, where results show that our idea is especially effective on CIFAR-10 and Kuzushiji-MNIST datasets.




(a) MNIST, $q = 0.1$. (b) Fashion-MNIST, $q = 0.3$.

(c) CIFAR-10, $q = 0.3$. (d) Kuzushiji-MNIST, $q = 0.5$.

*Figure 2.* Ablation study: comparisons between LW loss with losses on partial or non-partial labels.

#### 4.2.3. The Influence of Data Generation

In the data generation of previous subsections, the untrue partial labels are selected with equal probabilities, i.e. $q_z = q$ for $z \neq y$. In reality, however, some labels may be more analogous to the true label than others, and thus the probabilities $q_z$ for these labels may naturally be higher than others. In this part, we conduct empirical comparisons on data with alternative generation process. To be specific, Case 1 describes a "pairwise" partial label set, where there exists only one potential partial label for each class. In Case 2, we assume two potential partial labels for each class. Case 3 considers a more complex situation where 6 potential labels have different probabilities to enter the partial label set. More details about the data generations are shown in Section B.4 in the supplementary material. Other experimental settings are similar to Section 4.1.1.

As is shown in Table 3, our proposed method dominates its counterparts in all three cases. Moreover, as the data generation process becomes more complex (from Case 1 to Case 3), there is a natural drop in accuracy for all methods. Nonetheless, our LW-Cross entropy shows stronger resistance. For example, on Kuzushiji-MNIST, in Case

*Table 3.* Accuracy comparisons with different data generation.

| Dataset | Method | Base Model | Case 1 | Case 2 | Case 3 |
|---|---|---|---|---|---|
| MNIST | RC | MLP | $98.49 \pm 0.05\%*$ | $98.53 \pm 0.08\%*$ | $98.43 \pm 0.03\%*$ |
| | CC | MLP | $98.55 \pm 0.04\%*$ | $98.57 \pm 0.08\%*$ | $98.44 \pm 0.02\%*$ |
| | PRODEN | MLP | $98.64 \pm 0.15\%*$ | $97.61 \pm 0.10\%*$ | $98.55 \pm 0.12\%*$ |
| | LW-Sigmoid | MLP | $\underline{98.83 \pm 0.04\%}$ | $\mathbf{98.92 \pm 0.04\%}$ | $\underline{98.69 \pm 0.11\%}$ |
| | LW-Cross entropy | MLP | $\mathbf{98.88 \pm 0.05\%}$ | $\underline{98.88 \pm 0.09\%}$ | $\mathbf{98.82 \pm 0.05\%}$ |
| Kuzushiji-MNIST | RC | MLP | $92.61 \pm 0.17\%*$ | $92.47 \pm 0.19\%*$ | $92.07 \pm 0.10\%*$ |
| | CC | MLP | $92.65 \pm 0.15\%*$ | $92.68 \pm 0.10\%*$ | $91.91 \pm 0.15\%*$ |
| | PRODEN | MLP | $93.33 \pm 0.20\%*$ | $93.48 \pm 0.33\%*$ | $92.30 \pm 0.15\%*$ |
| | LW-Sigmoid | MLP | $\underline{93.80 \pm 0.15\%}$ | $\underline{93.87 \pm 0.14\%}*$ | $\underline{93.09 \pm 0.19\%}*$ |
| | LW-Cross entropy | MLP | $\mathbf{94.03 \pm 0.09\%}$ | $\mathbf{94.23 \pm 0.08\%}$ | $\mathbf{93.55 \pm 0.10\%}$ |
| Fashion-MNIST | RC | MLP | $89.79 \pm 0.10\%*$ | $89.88 \pm 0.11\%*$ | $89.47 \pm 0.11\%*$ |
| | CC | MLP | $89.63 \pm 0.12\%*$ | $89.58 \pm 0.20\%*$ | $88.63 \pm 0.33\%*$ |
| | PRODEN | MLP | $\underline{90.34 \pm 0.19\%}*$ | $89.88 \pm 0.27\%*$ | $89.60 \pm 0.14\%*$ |
| | LW-Sigmoid | MLP | $90.24 \pm 0.04\%*$ | $\underline{90.32 \pm 0.18\%}$ | $\underline{89.69 \pm 0.21\%}*$ |
| | LW-Cross entropy | MLP | $\mathbf{90.59 \pm 0.19\%}$ | $\mathbf{90.36 \pm 0.15\%}$ | $\mathbf{90.13 \pm 0.11\%}$ |
| CIFAR-10 | RC | ConvNet | $86.59 \pm 0.34\%*$ | $87.26 \pm 0.06\%*$ | $86.28 \pm 0.17\%*$ |
| | CC | ConvNet | $86.45 \pm 0.34\%*$ | $86.87 \pm 0.14\%*$ | $84.63 \pm 0.40\%*$ |
| | PRODEN | ConvNet | $89.03 \pm 0.59\%*$ | $88.19 \pm 0.10\%*$ | $87.16 \pm 0.13\%*$ |
| | LW-Sigmoid | ConvNet | $\mathbf{90.89 \pm 0.10\%}$ | $\mathbf{90.87 \pm 0.11\%}$ | $\underline{89.26 \pm 0.19\%}*$ |
| | LW-Cross entropy | ConvNet | $\underline{90.63 \pm 0.08\%}*$ | $\underline{90.51 \pm 0.14\%}*$ | $\mathbf{89.60 \pm 0.09\%}$ |

* The best results are marked in **bold** and the second best marked in underline. The standard deviation is also reported. We use $*$ to represent that the best method is significantly better than the other compared methods.

1 the accuracy of our LW-Cross entropy is $0.71\%$ higher than PRODEN, while in Case 3 the difference increases to $1.25\%$.

## 5. Conclusion

In this paper, we propose a family of loss functions, named *Leveraged Weighted* (LW) loss function, to address the problem of learning with partial labels. On the one hand, we provide theoretical guidance to the empirical choice of the leverage parameter $\beta$ proposed in our LW loss from the perspective of risk consistency. Both theoretical interpretations and empirical understandings show that $\beta = 1$ and $\beta = 2$ are preferred parameter settings. On the other hand, we design a practical algorithmic implementation of our LW loss, where its experimental comparisons with other state-of-the-art algorithms on both benchmark and real datasets demonstrate the effectiveness of our proposed method.

## Acknowledgements

Yisen Wang is supported by the National Natural Science Foundation of China under Grant No. 62006153, CCF-Baidu Open Fund (No. OF2020002), and Project 2020BD006 supported by PKU-Baidu Fund. Zhouchen Lin is supported by the National Natural Science Foundation of China (Grant No.s 61625301 and 61731018), Project 2020BD006 supported by PKU-Baidu Fund, Major Scientific Research Project of Zhejiang Lab (Grant No.s 2019KB0AC01 and 2019KB0AB02), and Beijing Academy of Artificial Intelligence.

## References

Allwein, E. L., Schapire, R. E., and Singer, Y. Reducing multiclass to binary: A unifying approach for margin classifiers. *Journal of Machine Learning Research*, 1(Dec):113–141, 2000.

Briggs, F., Fern, X. Z., and Raich, R. Rank-loss support instance machines for miml instance annotation. In *KDD*, 2012.

Cabannes, V., Rudi, A., and Bach, F. Structured prediction with partial labelling through the infimum loss. *arXiv preprint arXiv:2003.00920*, 2020.

Chen, C.-H., Patel, V. M., and Chellappa, R. Learning from ambiguously labeled face images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(7):1653–1667, 2017.

Clanuwat, T., Bober-Irizar, M., Kitamoto, A., Lamb, A., Yamamoto, K., and Ha, D. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

Cour, T., Sapp, B., Jordan, C., and Taskar, B. Learning from ambiguously labeled images. In *CVPR*, 2009.

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

Crammer, K. and Singer, Y. On the algorithmic implementation of multiclass kernel-based vector machines. *The Journal of Machine Learning Research*, 2(Dec):265–292, 2001.

Dempster, A. P., Laird, N. M., and Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society: Series B (Methodological)*, 39(1): 1–22, 1977.

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

Feng, L., Kaneko, T., Han, B., Niu, G., An, B., and Sugiyama, M. Learning with multiple complementary labels. In *ICML*, 2020a.

Feng, L., Lv, J., Han, B., Xu, M., Niu, G., Geng, X., An, B., and Sugiyama, M. Provably consistent partial-label learning. *arXiv preprint arXiv:2007.08929*, 2020b.

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, 2017.

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

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

Ishida, T., Niu, G., Hu, W., and Sugiyama, M. Learning from complementary labels. In *NeurIPS*, 2017.

Ishida, T., Niu, G., Menon, A., and Sugiyama, M. Complementary-label learning for arbitrary losses and models. In *ICML*, 2019.

Jin, R. and Ghahramani, Z. Learning with multiple labels. In *NeurIPS*, 2002.

Krizhevsky, A., Hinton, G., et al. Learning multiple layers of features from tiny images. *CiteSeer*, 2009.

Laine, S. and Aila, T. Temporal ensembling for semi-supervised learning. *arXiv preprint arXiv:1610.02242*, 2016.

LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Lee, Y., Lin, Y., and Wahba, G. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465):67–81, 2004.

Liu, L. and Dietterich, T. Learnability of the superset label learning problem. In *ICML*, 2014.

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

Luo, J. and Orabona, F. Learning from candidate labeling sets. In *NeurIPS*, 2010.

Lv, J., Xu, M., Feng, L., Niu, G., Geng, X., and Sugiyama, M. Progressive identification of true labels for partial-label learning. In *ICML*, 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*, 2019.

Nguyen, N. and Caruana, R. Classification with partial labels. In *KDD*, 2008.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

Rifkin, R. and Klautau, A. In defense of one-vs-all classification. *The Journal of Machine Learning Research*, 5:101–141, 2004.

Robbins, H. and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, pp. 400–407, 1951.

Tewari, A. and Bartlett, P. L. On the consistency of multiclass classification methods. In *COLT*. Springer, 2005.

Tewari, A. and Bartlett, P. L. On the consistency of multiclass classification methods. *Journal of Machine Learning Research*, 8(5), 2007.

Wang, D.-B., Li, L., and Zhang, M.-L. Adaptive graph guided disambiguation for partial label learning. In *KDD*, 2019.

Wang, W. and Zhang, M.-L. Semi-supervised partial label learning via confidence-rated margin maximization. In *NeurIPS*, 2020.

Wilcoxon, F. Individual comparisons by ranking methods. In *Breakthroughs in Statistics*, pp. 196–202. Springer, 1992.

Wu, X. and Zhang, M.-L. Towards enabling binary decomposition for partial label learning. In *IJCAI*, 2018.

Xia, X., Liu, T., Wang, N., Han, B., Gong, C., Niu, G., and Sugiyama, M. Are anchor points really indispensable in label-noise learning? In *NeurIPS*, 2019.

Xiao, H., Rasul, K., and Vollgraf, R. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

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

Yao, Y., Gong, C., Deng, J., and Yang, J. Network cooperation with progressive disambiguation for partial label learning. *arXiv preprint arXiv:2002.11919*, 2020.

Yu, F. and Zhang, M.-L. Maximum margin partial label learning. In *ACML*, 2016.

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

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

Zhang, M.-L., Zhou, B.-B., and Liu, X.-Y. Partial label learning via feature-aware disambiguation. In *KDD*, 2016.

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

Zhang, T. Statistical analysis of some multi-category large margin classification methods. *The Journal of Machine Learning Research*, 5(Oct):1225–1251, 2004.