Does Label Smoothing Help Deep Partial Label Learning?

Xiuwen Gong [1] **Nitin Bisht** [1] **Guandong Xu** [1 2]

Abstract

Although deep partial label learning (deep PLL) classifiers have shown their competitive performance, they are heavily influenced by the noisy false-positive labels leading to poorer performance as the training progresses. Meanwhile, existing deep PLL research lacks theoretical guarantee on the analysis of correlation between label noise (or ambiguity degree) and classification performance. This paper addresses the above limitations with label smoothing (LS) from both theoretical and empirical aspects. In theory, we prove lower and upper bounds of the expected risk to show that label smoothing can help deep PLL. We further derive the optimal smoothing rate to investigate the conditions, i.e., when label smoothing benefits deep PLL. In practice, we design a benchmark solution and a novel optimization algorithm called Label Smoothing-based Partial Label Learning (LS-PLL). Extensive experimental results on benchmark PLL datasets and various deep architectures validate that label smoothing does help deep PLL in improving classification performance and learning distinguishable representations, and the best results can be achieved when the empirical smoothing rate approximately approaches the optimal smoothing rate in theoretical findings. Code is publicly available at https://github.com/kalpiree/LS-PLL.

1. Introduction

Partial label learning (PLL) (Cour et al., 2011; Chen et al., 2014; Yu & Zhang, 2017) is an important weakly-supervised learning problem that allows each instance to be annotated with a set of candidate labels, with only one being the true label. PLL has attracted increasing attention in real-world applications due to its lower labeling cost, such as, automatic face recognition, automatic object detection, etc.

In recent years, PLL has evolved from conventional PLL to deep PLL (i.e., from linear/kernel-based models to deep neural networks (DNNs)-based models) as a result of DNNs' remarkable training ability on large-scale datasets. The main challenge for deep PLL is that the ground-truth label in PLL is unkown while DNNs largely depend on the precisely labeled data to guarantee the effectiveness of training. More specifically, DNNs are over-confident on any fed example. If the DNNs are not fed data with true labels, the learned deep PLL classifiers perform worse than the desired ones as with the training process continues. Moreover, what changes with the PLL evolvement is the label noise, from uniform to non-uniform, and from naive to competitive/realistic. Most deep PLL methods assume that the noisy labels are randomly generated from a uniform distribution (Lv et al., 2020; Feng et al., 2020; Wen et al., 2021; Wang et al., 2022). However, the noisy labels are usually high-correlated with the true label and instances in real-world scenarios, which makes the deep PLL problem more challenging than the uniformally generated label noise. As is also pointed out by Xu et al. (2021) that the candidate labels are always instance-dependent. Yan & Guo (2023) also point out that it is more realistic to consider the competitive noise, which can demonstrate stronger association relationships with a true label than a random label noise.

Here comes a question: What will happen when competitive noise meets the over-confidence of deep PLL? Larger performance degradation, definitely! In addition, existing research on deep PLL lacks theoretical guarantee on the analysis of correlation between label noise (or ambiguity degree) and classification performance. These problems motivate us to figure out a solution. Label smoothing (Szegedy et al., 2016) has shown to be effective in denoising and improving predictive performance of deep learning models by preventing DNN from becoming over-confident. Does label smoothing help deep PLL? If so, when can label smoothing benefit PLL? Little research has been done to explore this.

In this paper, we provide a novel insight into deep PLL by investigating the effectiveness of label smoothing from both theoretical and empirical aspects. We first propose a novel

[1] Faculty of Engineering and Information Technology, University of Technology Sydney, NSW, Australia [2] Department of Computing, The Hong Kong Polytechnic University (PolyU), Kowloon, Hong Kong. Correspondence to: Guandong Xu <Gdxu@eduhk.hk>.

Proceedings of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

label smoothing-based expected risk for deep PLL. Then, we put forward a definition called generalized ambiguity degree to quantify the competitive label noise in real-world scenarios. Theoretically, we show that label smoothing can help deep PLL under mild conditions. We prove lower and upper bounds of the expected risk w.r.t. label smoothing on partial data to show its approximate equivalence to the expected risk w.r.t. the label smoothing on clean data. Moreover, we derive the optimal smoothing rate to investigate the conditions. We pioneer the theorectical guarantee on the correlation between the classification performance (determined by the optimal smoothing rate) and label noise (quantified by the generalized ambiguity degree), which sheds light on the parameter choice of smoothing rate in empirical studies. We also prove an estimation error bound of the empirical risk to further validate the effectiveness of label smoothing for deep PLL. Empirically, we propose a benchmark solution, and design a novel algorithm named Label Smoothing-based Partial Label Learning (LS-PLL) to optimize the objective function with respect to the empirical risk. Extensive experimental results on various competitive noise PLL datasets and different deep architectures validate that label smoothing does help deep PLL in improving performance and learning distinguishable representations, and the best results can be achieved when the empirical smoothing rate approximately approaches the optimal smoothing rate in theoretical findings.

2. Related Work

Partial label learning (PLL), also known as ambiguous-label learning (Hüllermeier & Beringer, 2006; Zeng et al., 2013) or superset-label learning (Gong et al., 2018; Liu & Dietterich, 2014; 2012), is a weakly supervised learning problem (Zhou, 2017), which differs from (semi-)supervised learning (Liu & Tsang, 2017; Liu et al., 2017; 2019; Chen et al., 2023; Mao et al., 2023). In PLL, each instance has a collection of candidate labels, only one of which is the ground-truth label while the others are false positive labels, resulting in ambiguity while training classification models.

Conventional partial-label learning. Conventional disambiguation methods for PLL can be broadly divided into two categories (Lyu et al., 2021; Zhou et al., 2017): disambiguation by candidate label average methods, or disambiguation by ground-truth label identification methods. For the average-based methods (Cour et al., 2011; Hüllermeier & Beringer, 2006; Zhang & Yu, 2015), 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, these 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 (Liu & Dietterich, 2012; Yu & Zhang, 2017; Chai et al., 2020), the ground-truth label is regarded as a latent variable and identified through an iterative refining procedure. In addition, some labeling confidence-based approaches are proposed. 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. 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. 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. With a high demand of big data in real-world applications, conventional PLL methods has shown limitations on processing large-scale datasets.

Deep partial-label learning. Recently, neural network based methods are widely developped for PLL due to their remarkable training ability on large-scale datasets. For example, 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. Yao et al. (2020) proposed a PLL method called NCPD to train two networks in a mutual learning manner in order to alleviate the error accumulation problem. Feng et al. (2020) proposed a generation model of candidate label sets, and develop two novel PLL methods (i.e., RC and CC) that are guaranteed to be provably consistent. Wen et al. (2021) proposed a family of loss functions, which achieves risk consistency by generalizing the uniform assumption on the generation procedure of partial label sets. Wang et al. (2022) proposed a contrastive learning-based method by embedding class prototype-based label disambiguation strategy. Recently, Yan & Guo (2023) proposed a mutual learning-based method to solve the PLL problem, which is the first work that considers the competitive noise setting under deep learning framework, and achieves SOTA prediction performance for PLL.

Although deep PLL has shown its competitive performance, it is still negatively and heavily impacted by the noisy false-positive labels. This is because DNNs are largely dependent on the precisely labeled data to guarantee effectiveness of training, and are over-confident on any fed example. Specifically, if the DNNs are not fed data with true labels, the learned deep PLL classifiers would be misled by the false positive candidate labels and perform worse than the desired ones as with the training process going on. What will happen when the competitve noise meets the over-confidence of deep PLL? Larger performance degradation, definitely! This motivates us to think of a way to alleviate this problem.

Label smoothing. Label smoothing is an emerging learning paradigm, which has shown promise in denoising and also in improving predictive performance of deep learning models by preventing DNN from becoming over-confident (Szegedy et al., 2016; Müller et al., 2019; Lukasik et al., 2020; Wei et al., 2022). For any instance $\boldsymbol{x}$, let $\boldsymbol{y}$ be the one-hot vector form of the ground-truth label y, and $\boldsymbol{y}^{LS}$ be the label smoothing form of $\boldsymbol{y}$. Following Szegedy et al. (2016), the label smoothing formulation is defined as below:

$$\boldsymbol{y}^{LS} = (1-r) \cdot \boldsymbol{y} + \frac{r}{L} \cdot \mathbf{1} \tag{1}$$

where $r \in [0,1]$ is the smoothing rate; L is the total number of classes; and $\mathbf{1}$ is the all-ones vector. In above label smoothing definition, one mixes the training label with a uniform mixture over all possible labels. For example, when $r = 0.4$, the smoothed label of $\boldsymbol{y} = [0,1,0,0]^T$ becomes $\boldsymbol{y}^{LS} = [0.1, 0.7, 0.1, 0.1]^T$.

Does label smoothing help deep PLL? If so, when does label smoothing benefit PLL? These research has been unexplored. This paper will present a novel perspective by connecting label smoothing to deep PLL with solid theorectical gurantees and empirical studies.

3. Theoretical Analysis

In this section, we provide a closer look at partial label learning in view of Label Smoothing (LS). Firstly, we develop definitions of smoothed partial label and the smoothed partial label loss, and then propose the expected risk based on the defined loss. Secondly, we define generalized ambiguity degree for PLL, which is crucial important for later theorems and proofs. Thirdly, we prove lower and upper bounds for the proposed risk and derive the optimal smoothing rate through analysis. Last but not the least, an estimation error bound of empirical risk is proved.

To start with, we define some notations in the paper. Let $\mathcal{S} = \{(\boldsymbol{x}_i, Y_i)_{i=1}^n\} = \{(\boldsymbol{x}_i, \{y_i\} \cup Z_i)_{i=1}^n\}$ denote the partial label dataset drawn i.i.d. n times from a distribution $\mathbb{P}$. For each training example $(\boldsymbol{x}_i, Y_i)$, we have an instance $\boldsymbol{x}_i \in \mathbb{R}^d$ with d features, and a corresponding candidate label set $Y_i \subseteq \mathcal{Y}$, where $\mathcal{Y} = [L] \doteq \{1, \cdots, L\}$. In addition, let y_i denote the ground-truth label of x_i, which is known to reside in the corresponding candidate label set, but cannot be directly accessible in the training phase; let Z_i denote the set of false-positive candidate labels. With the ablove definitions, we have $Y_i = \{y_i\} \cup Z_i$, $y_i \in Y_i$, $Z_i \subset Y_i$, $y_i \notin Z_i$. We further assume that the clean date $(\boldsymbol{x}_i, y_i)$ is drawn from some unknown distribution $\mathbb{D}$. Let $\mathbf{X}$, $\mathbf{Y}$, $\mathbf{Y}^*$, $\mathbf{Z}$ be the random variables of x_i, Y_i, y_i, Z_i respectively. We then have $(\mathbf{X}, \mathbf{Y}) \sim \mathbb{P}$, $(\mathbf{X}, \mathbf{Y}^*) \sim \mathbb{D}$. To simplify formulations, we use $(\mathbf{X}, \mathbf{Y})$ to denote $(\mathbf{X}, \mathbf{Y}) \sim \mathbb{P}$, and use $(\mathbf{X}, \mathbf{Y}^*)$ to denote $(\mathbf{X}, \mathbf{Y}^*) \sim \mathbb{D}$ in the expectations.

3.1. The Expected Risk

We first derive the expected risk for label smoothing-based PLL, and then provide theoretical analysis of the risk.

3.1.1. Derivation of The Expected Risk

In PLL, the ground-truth label is unknown, but is assumed to reside in the candidate label set. This makes label smoothing for PLL to be different from the traditional definition in Eq. (1) in Appendix. To cope with this, we first define smoothed partial labels for PLL as follows:

Definition 3.1 (Smoothed Partial Labels). Let (x, Y) be a training example, where x denotes the instance and Y denotes the candidate label set. Let $\boldsymbol{Y} \in \{0,1\}^L$ denote the corresponding L-dimension label vector of Y; $\boldsymbol{Y}^{LS} \in \mathbb{R}^L$ denote the smoothed label vector of $\boldsymbol{Y}$; $Y^{LS,j}$ be the j-th element of $\boldsymbol{Y}^{LS}$. Assume that y is the ground-truth label of the instance x, we then have the smoothed labels of the training example as follows:

$$Y^{LS,j} = \begin{cases} (1-r) \cdot \mathbb{I}_{j=y} + \frac{r}{|Y|}, & \text{if } j \in Y \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

where $\mathbb{I}$ is an indicator function; r is the smoothing rate, and $|Y|$ is the size of candidate label set.

Under the definition of label smoothing for PLL, we mix the training labels with a uniform mixture over all candidate labels. For example, when $r = 0.2$, $y = 3$, the smoothed label vector of $\boldsymbol{Y} = [0,1,1,0]^T$ becomes $\boldsymbol{Y}^{LS} = [0, 0.1, 0.9, 0]^T$.

We use f to denote a deep neural network, and $\mathbf{f}(x_i) \in \mathbb{R}^L$ denotes the model prediction scores of x_i, with $f^j(x_i)$ being the j-th element. In the following parts, we use f^j to denote $f^j(x_i)$ for simplicity. Let $\ell : [L] \times \mathbb{R}^L \to \mathbb{R}_+$ be a loss function, where $\ell(\mathbf{f}(x_i), y_i)$ is the penalty for the model prediction scores given the true label $y_i \in Y_i$. Throughout this paper, we use softmax cross-entropy loss (SCE) as the loss function ℓ. When SCE is applied to the hard labels in multiclass classification, $\ell(\mathbf{f}(x), y) = -f^y + \log \sum_{k \in [L]} e^{f^k}$ for any training example (x, y), where y is the correct class. However, the ground-truth label is unknown in PLL, thus we cannot apply multi-class SCE directly to PLL. Instead, we choose to minimize SCE between the smoothed candidate labels and the soft outputs of the deep neural networks. Following the definition of smoothed partial labels in Definition 3.1, we can further define the loss function of smoothed partial labels as follows:

Definition 3.2 (Smoothed Partial Label Loss)**.** For any training example (x, Y), the softmax cross-entropy loss for label smoothing-based PLL can be defined as follows:

$$\ell(\mathbf{f}(x), \mathbf{Y}^{LS}) = \sum_{j \in Y} Y^{LS,j}\big(- f^j + \log \sum_{k \in [L]} e^{f^k}\big) \tag{3}$$

where $Y^{LS,j}$ is defined Eq. (2).

To this end, we can define the expected risk for the label smoothing-based PLL with the following formulation:

$$\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\ell\big(\mathbf{f}(\mathbf{X}), \mathbf{Y}^{LS}\big)\big] \tag{4}$$

3.1.2. Analysis of Expected Risk

In this subsection, we aim to analyze the composition underlying the expected risk via the following thereom.

Theorem 3.3. *The expected risk minimization with respect to (w.r.t.) smoothed partial labels $\mathbf{Y}^{LS}$ in PLL setting (Eq. (4)) is equivalent to the expected risk w.r.t. the unobserved correct label $\mathbf{Y}^*$ defined on the clean data, and the expected risk w.r.t. the candidate labels $\mathbf{Y}$ defined on the observed partially labeled data:*

$$\begin{aligned} &\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\ell\big(\mathbf{f}(\mathbf{X}), \mathbf{Y}^{LS}\big)\big] \\ &= \mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[(1-r)\ell\big(\mathbf{f}(\mathbf{X}), \mathbf{Y}^*\big)\big] \\ &\quad + \mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\frac{r}{|\mathbf{Y}|}\sum_{j \in \mathbf{Y}} \ell\big(\mathbf{f}(\mathbf{X}), j\big)\big] \end{aligned} \tag{5}$$

Proof. Proof of this theorem can be found in Appendix. □

Remark. Since our goal is to minimize the expected risk in Eq. (4) (i.e., the term on the left side of Theorem 3.3), we need that the sum of both terms on the right side of Theorem 3.3 to be small. When the smoothing rate $r \to 0$, we can use the expected risk on smoothed partial label data to approximately estimate the expected risk w.r.t. the clean data. When the smoothing rate $r \to 1$, we can use the expected risk on smoothed partial label data to approximately estimate the expected risk w.r.t. the observed partial data. When $r \in (0, 1)$, the expected risk w.r.t. smoothed partial labels is equivalent to the risks trading off between the risk on clean data and that on the observed partial data. To sum up, the proposed expected risk w.r.t. smoothed partial labels in PLL setting (Eq. (4)) could be taken as an effective way for partial label learning according to the choice of r. Intuitively, the label smoothing rate r is supposed to be correlated with the noise levels or the ambiguity degree of the partially labeled data. Next, we define a crucial term, called the generalized ambiguity degree, that will be used in later theorems and derivation of the optimal smoothing rate.

3.2. Generalized Ambiguity Degree

We define the generalized ambiguity degree ϵ of partial label learning (PLL) w.r.t. distribution $P(\mathbf{X}, \mathbf{Y}^*, \mathbf{Z})$ as follows:

Definition 3.4 (Generalized Ambiguity Degree of PLL)**.**

$$\epsilon = \sup_{Z \subseteq \mathcal{Y} \setminus y} P(\mathbf{Z} = Z | \mathbf{X} = x, \mathbf{Y}^* = y) \tag{6}$$

In words, the above-defined ambiguity degree ϵ corresponds to the maximum probability of all candidate labels co-occurring with the ground-truth label in PLL. Here, $Z = Y \setminus y, Y \subseteq \mathcal{Y}$ and $|Z| \geq 1$. When $\epsilon = 0$, we have $Z = \varnothing$, then the candidate label set only includes the ground-truth label, which becomes the traditional multi-class classification problem. When $\epsilon = 1$, we have $Z = \mathcal{Y} \setminus y$, then the candidate label set includes all the labels, which turns out to be the traditional unsupervised learning problem. Thus, we only consider the ambiguity degree between 0 and 1 for PLL, i.e., $0 < \epsilon < 1$.

Different from the ambiguity degree defined by Cour et al. (2011), our definition can be taken as a generalization form. Specifically, our generalized ambiguity degree corresponds to the maximum probability of all false-positive candidate labels co-occurring with the ground-truth label, while Cour et al. (2011) considers only one false-positive candidate label. In real-world scenarios, candidate label set often involves more than one false-positive labels that are chosen by annotators according to random or competitive strategies. All these false-positive labels are supposed to have negative impact on disambiguation, and the extent of difficulty in disambiguation depends on the extent of ambiguity. Thus, it is more reasonable to consider the generalized ambiguity degree in real-world scenarios. For example, in online image annotation, the true label of a dog image is Alaskan Malamute, but online annotators with limited professional knowledge may provide the image with downgraded label confidence like Alaskan Malamute (0.30), Siberian Husky (0.10), German Shepherd (0.10), Border Collie (0.05), Setters (0.04), Irish Wolfhound(0.03), and many other labels with lower confidence. If the size of candidate label set is required to be 3, the unconfident annotator may provide the candidate label set as {Alaskan Malamute (0.30), Siberian Husky (0.10), German Shepherd (0.10)}, {Alaskan Malamute (0.30), Siberian Husky (0.10), Border Collie (0.05)}, or {Alaskan Malamute (0.30), Siberian Husky (0.10), Setters (0.04)}. Assume candidate labels are i.i.d. generated, then the ambiguity degree given our definition will be $0.20 = \sup\{0.20, 0.15, 0.14\}$, while Cour et al. (2011) only considers the highest probability of one candidate label, the ambiguity degree is $0.10 = \sup\{0.10, 0.10, 0.10\}$. Our definition of ambiguity degree can demonstrate the difference among the three candidate label sets, while Cour et al. (2011) can not. Thus, our definition of ambiguity degree is more suitable for the competitive noise in real-world applications, and the uniformly generated noise could be taken as a special case with all values being equal.

3.3. Lower and Upper Bounds

We prove the lower and upper bounds of the expected risk, and further derive the optimal smoothing rate r_{opt}. To start with, we provide the lower bound in the following theorem:

Theorem 3.5 (Lower bound)**.** *For any instance $(\boldsymbol{x}, \{y\}\cup Z)$, let $|Z| = c$, r_{opt} denote the optimal smoothing rate on partial data. r^* denotes the optimal smoothing rate on clean data, and is usually set to a very small value in empirical practice. Suppose $t_1 = \left(1 - r + \frac{r}{c+1}\right)\frac{1}{1-r^*}$, $t_2 = \frac{t_1}{\binom{L}{c}} \cdot \frac{r^*}{L}$, $t_3 = \left(\frac{r\epsilon}{c+1} - t_2\right)$, then the expected risk w.r.t. the loss of (unobserved) smoothed correct label $\mathbf{Y}^{*LS}$ defined on clean multi-class data could be lower bounded by the expected risk w.r.t. the loss of (observed) smoothed partial label $\mathbf{Y}^{LS}$ defined on PLL data with two extra bias terms as follows:*

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[\ell\big(\mathbf{f}(\mathbf{X}), \mathbf{Y}^{*LS}\big)\big] \\
&\geq \mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\frac{1}{t_1}\ell\big(\mathbf{f}(\mathbf{X}), \mathbf{Y}^{LS}\big)\big] \\
&\quad + \mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[\frac{t_2}{t_1}\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}\sum_{j\in\bar{Z}}\ell\big(\mathbf{f}(\mathbf{X}), j\big)\big] \\
&\quad - \mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[\frac{t_3}{t_1}\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}\sum_{j\in Z}\ell\big(\mathbf{f}(\mathbf{X}), j\big)\big]
\end{aligned} \tag{7}$$

Proof. Proof of this theorem can be found in Appendix. □

It is known that $t_1 > 0$ as a result of smoothing rate $r \in [0, 1]$, and we can then get $t_2 > 0$. Consequently, the lower bound in (7) will become tighter by making the weight of the third term equal to zero, i.e., $t_3 = 0$. With some calculations, we can derive the *optimal partial label smoothing rate r_{opt}*:

$$r_{opt} = \frac{(c+1)r^*}{\Delta}, \text{where } \Delta = L\binom{L}{c}\epsilon(1-r^*) + cr^* \tag{8}$$

To this end, we bridge a connection between the optimal label smoothing rate and the generalized ambiguity degree by Eq. (8). By applying the optimal smooth rate r_{opt} to Eq. (7), we further derive an optimized lower bound.

Theorem 3.6 (Optimized lower bound)**.** *Under the conditions in Theorem 3.5, partial label learning with the optimal smoothing rate $r = \frac{(c+1)r^*}{\Delta}$, where $\Delta = L\binom{L}{c}\epsilon(1-r^*) + cr^*$, yields the optimal lower bound as follows:*

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[\ell\big(\mathbf{f}(\mathbf{X}), \mathbf{Y}^{*LS}\big)\big] \\
&\geq \mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\frac{1}{t_1}\cdot\ell\big(\mathbf{f}(\mathbf{X}), \mathbf{Y}^{LS}\big)\big] \\
&\quad + \mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[\frac{t_2}{t_1}\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}\sum_{j\in\bar{Z}}\ell\big(\mathbf{f}(\mathbf{X}), j\big)\big]
\end{aligned} \tag{9}$$

Next, we demonstrate the upper bound as follows:

Theorem 3.7 (Upper bound)**.** *Suppose the ambiguity degree satisfies the condition that $\epsilon \in (\delta, 1)$ and $\delta = \frac{Lc-(L-1)cr^*}{L(L-1)\binom{L}{c}(1-r^*)}$, the expected risk w.r.t. the loss of (unobserved) smoothed correct label $\mathbf{Y}^{*LS}$ defined on clean multi-class data could be lower bounded by the expected risk w.r.t. the loss of (observed) smoothed partial label $\mathbf{Y}^{LS}$ defined on PLL data with a bias term as follows:*

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[\ell\big(\mathbf{f}(\mathbf{X}), \mathbf{Y}^{*LS}\big)\big] \\
&\leq \mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\ell\big(\mathbf{f}(\mathbf{X}), \mathbf{Y}^{LS}\big)\big] \\
&\quad + \mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\frac{r^*}{L}\sum_{j\in\bar{\mathbf{Y}}}\ell\big(\mathbf{f}(\mathbf{X}), j\big)\big]
\end{aligned} \tag{10}$$

Proof. Proof of this theorem can be found in Appendix. □

Remark. Theorem 3.6 and Theorem 3.7 suggests that the expected risk w.r.t. (unobserved) smoothed correct label loss defined on the clean multi-class data could be lower and upper bounded by the expected risk w.r.t. (observed) smoothed partial label loss defined on PLL data. When the partial label smoothing rate r is chosen to be optimal in Theorem 3.6 and the ambiguity degree ϵ satisfies the condition in Theorem 3.7, the optimization of smoothed clean risk (unobserved) could be approximately estimated by that of the smoothed partial risk (observed), which theoretically demonstrates our thinking of whether and when label smoothing can help deep PLL.

3.4. Estimation Error Bound

In this subsection, we prove an estimation error bound for the empirical risk to further demonstrate its effectiveness theoretically. To begin with, we restate some basic definitions, i.e., Rademacher complexity (Bartlett & Mendelson, 2002) and ρ-Lipschitz (Shalev-Shwartz & Ben-David, 2014), in Definiton A.1 and Definiton A.2 in Appendix. Given the definitions, we can derive the following Lemma:

Lemma 3.8 (ρ-Lipschitz for Label Smoothing-based PLL Loss)**.** *Assume that the softmax cross-entropy loss (SCE) $\ell\big(\mathbf{f}(x), y\big)$ is ρ-Lipschitz for every y, then the proposed Label Smoothing-based PLL Loss $\ell\big(\mathbf{f}(x), \mathbf{Y}^{LS}\big)$ defined in equation (3) is also ρ-Lipschitz with respect to $\mathbf{f}(x)$ for all $y \in Y$.*

Proof. Proof of this lemma can be found in Appendix. □

Lemma 3.9. *Define a function space as* $\mathcal{G} = \{(x, Y) \mapsto \ell(\mathbf{f}(x), Y^{LS}) | \mathbf{f} \in \mathcal{F}\}$*, then the following inequality holds:*

$$\mathfrak{R}_n(\mathcal{G}) \leq \sqrt{2}\rho \sum_{y \in Y} \mathfrak{R}_n(\mathcal{F}_y)$$

where $\mathcal{F}_y \doteq \{\mathbf{f} : x \mapsto \mathbf{f}(x) | \mathbf{f} \in \mathcal{F}, y \in Y\}$, $\mathfrak{R}_n(\mathcal{F}) = \mathbb{E}_{\mathbf{X}}\mathbb{E}_{\sigma}[\sup_{f \in \mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \mathbf{f}(x_i)]$.

Proof. Proof of this lemma can be found in Appendix. □

Let $\mathcal{R}_{exp}(f)$ denote the expected risk for the label smoothing-based PLL defined in Eq. (4), and let $\mathcal{R}_{emp}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell\big(\mathbf{f}(x_i), Y_i^{LS}\big)$ denote the empirical risk. Given the Rademacher bound on the maximal deviation between the expected risk and the empirical risk, we have the following theorem:

Theorem 3.10 (Estimation Error Bound)**.** *Assume the label smoothing-based PLL loss function* $\ell(\mathbf{f}(x), Y^{LS})$ *is upper-bounded by* M*, i.e.,* $M = \sup_{\mathbf{f} \in \mathcal{F}} \ell(\mathbf{f}(x), Y^{LS})$*. For any* $\eta > 0$*, with a probability of at least* $1 - \eta$*, we have*

$$\begin{aligned} &\sup_{f \in \mathcal{F}} |\mathcal{R}_{exp}(f) - \mathcal{R}_{emp}(f)| \\ &\leq 2\sqrt{2}\rho \sum_{y \in Y} \mathfrak{R}_n(\mathcal{F}_y) + \frac{M}{2}\sqrt{\frac{\log \frac{2}{\eta}}{2n}} \end{aligned}$$

Proof. Proof of this theorem can be found in Appendix. □

Remark. Theorem 3.10 establishes a connection between the expected risk $\mathcal{R}_{exp}(f)$ and the empirical risk $\mathcal{R}_{emp}(f)$ with an upper bound. When the sample size $n \to \infty$, the empirical risk $\mathcal{R}_{emp}(f)$ is expected to provide a good approximation of the expected risk $\mathcal{R}_{exp}(f)$.

4. Benchmark Solution

In the previous section, our theories are based on the expected risk. In practice, we can only observe the finite PLL samples while the distribution of partial label datasets are unknown. In this section, we develop an optimization algorithm, called Label Smoothing-based Partial Label Learning Algorithm (LS-PLL Algorithm), to optimize the empirical risk by training basic neural architectures and identifying the ground-truth label alternately.

4.1. Neural Network Training

Let f denote a predictive network; let y be a latent ground-truth variable. We can get the objective function of LS-PLL given the empirical risk w.r.t. the smoothed partial data as follows:

$$\mathcal{L}(f, y) = \frac{1}{n}\sum_{i=1}^{n}\sum_{j \in Y_i} -Y_i^{LS,j}\left(f_i^j - \log \sum_{k \in [L]} e^{f_i^k}\right) \tag{11}$$

where $Y_i^{LS,j} = \begin{cases} (1-r)\cdot \mathbb{I}_{j=y_i} + \frac{r}{|Y_i|}, & \text{if } j \in Y_i \\ 0, & \text{otherwise} \end{cases}$.

In each training iteration, we perform label smoothing on the identified ground-truth label, and the ground-truth label at the warm-up stage is randomly chosen from the candidate label set. We then train the predictive network by minimizing the objective function in Eq. (11) with a mini-batch based stochastic gradient descent algorithm. As the performance of a prediction network is dependent on the identified ground-truth label, we provide the ground-truth label identification strategy in the next subsection.

4.2. Ground-truth Label Identification

For any instance $\boldsymbol{x}_i$, the probability vector of the predicton network's output is denoted by $\mathbf{f}(\boldsymbol{x}_i)$ with f_i^j being the j-th element; the corresponding probability vector of the softmax output is denoted by $\boldsymbol{q}_i$, with the j-th element being

$$q_{ij} = \frac{e^{f_i^j}}{\sum_{k \in [L]} e^{f_i^k}} \tag{12}$$

In order to make the effect of predicting the ground-truth label being stably accumulated with the increasing iterations of training process, we update q_{ij} in a moving-average manner as follows:

$$q_{ij}^t = \eta q_{ij}^t + (1-\eta) q_{ij}^{t-1} \tag{13}$$

where $\eta \in (0, 1)$ is a weighting parameter; t denotes the t-th iteration.To mitigate the negative impact of false-positive labels in the training process, we restrict $j \in Y_i$. To guarantee the effectiveness of the accumulated softmax score q_{ij}^t, we constrain it to be within $(0, 1)$ with the following normalization:

$$\hat{q}_{ij}^t = \begin{cases} \frac{q_{ij}^t}{\sum_{k \in Y_i} q_{ik}^t}, & \text{if } j \in Y_i \\ 0, & \text{otherwise} \end{cases} \tag{14}$$

Consequently, for each instance $\boldsymbol{x}_i$, we can identify its ground-truth label y_i at the t-th iteration as below:

$$y_i^t = \arg\max_j \hat{q}_{ij}^t \tag{15}$$

As the neural network training and ground-truth label identification alternatively proceed in an iteration manner, the

Algorithm 1 LS-PLL Algorithm

```
Input: PLL training data $\mathcal{S}$; smoothing rate $r$; trade-off weighting parameter $\eta$.
Initialize ground-truth label $y_i$ randomly chosen from $Y_i$ for each instance $x_i$;
for $iter = 1, 2, \ldots$ do
    for $batch = 1, 2, \ldots$ do
        Sample a mini-batch $B$ from $\mathcal{S}$;
        Calculate smoothed labels for each instance on current batch B according to Eq. (2);
        Update the predictive network by minimizing the objective function $\mathcal{L}$ in Eq. (11) with stochastic gradient descent (SGD) algorithm;
        Update the softmax output for each instance on current batch according to Eq. (13), and then calculate Eq. (14);
        Update the ground-truth label for each instance on current batch via Eq. (15).
    end for
end for
```

probability vector of the softmax outputs updated through moving-average and normalization strategies will become more reflective on the ground-truth label, which in turn brings a positive impact on the subsequent network training, and gradually reduces the negative impact of false-positive labels in the training process. The complete procedures for implementing the proposed LS-PLL algorithm on each mini-batch are summarized in Algorithm 1.

5. Experiments

We conduct experiments to validate the effectiveness of label smoothing for deep PLL from two aspects: **1)** validate 'whether' label smoothing is effective for deep PLL. **2)** vadidate 'when' label smoothing benefits PLL.

5.1. Datasets and Implementation Details

We conduct experiments on four commonly used benchmark datasets, i.e., Fashion-MNIST (Xiao et al., 2017), Kuzushiji-MNIST (Clanuwat et al., 2018), CIFAR-10 and CIFAR-100 (Krizhevsky, 2009). Similar to the generation procedures of candidate labels with competitive noise in Yan & Guo (2023), we first produce a new label space constituted by top-K highest probabilities predicted by a neural network trained on the original clean dataset excluding the ground-truth label. We then randomly generate integers as the size of candidate labels for each instance with some constraints: 1) no less than one, no greater than one plus the size of new label space; 2) the average number of all random integers equals to that of candidate labels (Avg. #CL) specified in our experiments. According to the size of candidate label set, we generate noisy labels by choosing from the new label space in a descending order of probability (excluding the true label) rather than randomly choosing, which aims to make the label noise more competitive considering that more similar labels result in higher disambiguity. The chosen noisy labels together with the true label form the candidate label set. In detail, we choose the new label space from the top-6 predictions for Fashion-MNIST, Kuzushiji-MNIST and CIFAR-10; and top-20 predictions for CIFAR-100. The average number of candidate labels (Avg. #CL) is specified to be 3, 4, 5 for Fashion-MNIST, Kuzushiji-MNIST and CIFAR-10 respectively, and 7, 9, 11 for CIFAR-100. We employ LeNet-5 as the neural architecture on the Fashion-MNIST and Kuzushiji-MNIST datasets, ResNet-18 on CIFAR-10, and ResNet-56 on CIFAR-100. The optimizer is stochastic gradient descent (SGD) (Robbins et al., 1951) with momentum 0.9 and a weight decay of $1e-3$ for model training. The mini-batch size, learning rate and total training epochs are set to 128, 0.01, and 200 respectively. Moreover, the empirical smoothing rate r is chosen from $\{0.1, 0.3, 0.5, 0.7, 0.9\}$. The weighting parameter η is set to be 0.9. The characteristics of each benchmark dataset (including the number of features, classes, Avg.#CL and architectures) are reported in Table 5 in Appendix.

5.2. Experimental Results w.r.t. 'Whether'

We conduct experiments to validate whether label smoothing is effective for deep PLL from two sides: Effect of label smoothing on classification performance; Effect of label smoothing on pre-logits.

Effect of label smoothing on classification performance: In order to demonstrate how the classification performance is influenced by label smoothing for deep PLL, we compare test accuracies of basic neural architectures with label smoothing (i.e., w/ LS) and without label smoothing (i.e., w/o LS) on PLL datasets under various noise levels ((i.e., Avg. #CL) in Table 1 and Table 2. From these results, we can observe that the prediction accuracy w/ LS (i.e., LS-PLL) significantly outperform those w/o LS on all PLL datasets under various noise levels. Meanwhile, the results are consistent across all architectures, which indicates that label smoothing is effective for deep PLL regardless of different neural network architectures. Moreover, the superiority of label smoothing is more evident as the varying noise level becomes larger on all datasets. For example, on CIFAR-10, LS-PLL improves the best accuracy w/o LS by approximately 19%, 26%, 29% respectively. It can thus, easily be concluded that label smoothing is effective for deep PLL with superior classification performance.

Effect of label smoothing on pre-logits in PLL: To illustrate how pre-logits (i.e., penultimate layer representations) are affected by label smoothing, we follow the visualization

Table 1. Test accuracy of training basic network architectures LeNet-5, ResNet-18 with and without label smoothing on Fashion-MNIST, Kuzushiji-MNIST, CIFAR-10 with various noise levels (i.e., Avg.#CL =3, 4, 5). The best results are highlighted in bold.

Dataset	Achitecture	Acc (Avg.#CL = 3)	Acc (Avg.#CL = 4)	Acc (Avg.#CL = 5)
Fashion-MNIST	LeNet-5 w/o LS	87.80	83.36	81.23
	w/ LS (i.e., LS-PLL)	**89.99**	**84.32**	**84.06**
Kuzushiji-MNIST	LeNet-5 w/o LS	85.28	76.31	67.25
	w/ LS (i.e., LS-PLL)	**90.64**	**87.85**	**76.10**
CIFAR-10	ResNet-18 w/o LS	63.39	51.47	38.96
	w/ LS (i.e., LS-PLL)	**82.97**	**77.68**	**67.23**

Table 2. Test accuracy of training basic network architecture ResNet-56 with and without label smoothing on CIFAR-100 with various noise levels (i.e., Avg.#CL =7, 9, 11). The best results are highlighted in bold.

Dataset	Method	Acc (Avg.#CL = 7)	Acc (Avg.#CL = 9)	Acc (Avg.#CL = 11)
CIFAR-100	ResNet-56 w/o LS	34.15	33.60	21.00
	w/ LS (i.e., LS-PLL)	**57.90**	**47.50**	**35.50**

Table 3. Test accuracy comparisons of LS-PLL w.r.t. varying smoothing rate r on Fashion-MNIST (FMNIST), Kuzushiji-MNIST(KMNIST) and CIFAR-10 datasets at different noise levels (i.e., Avg.#CL =3, 4, 5). The best results are highlighted in bold.

Dataset	SmoothRate	Avg.#CL = 3	Avg.#CL = 4	Avg.#CL = 5
FMNIST	r = 0.1	84.39	72.96	52.55
	r = 0.3	84.72	66.89	55.87
	r = 0.5	84.73	76.42	47.56
	r = 0.7	84.80	81.48	47.95
	r = 0.9	**89.99**	**84.32**	**84.06**
KMNIST	r = 0.1	81.11	47.99	36.10
	r = 0.3	88.22	53.39	33.68
	r = 0.5	89.34	55.17	50.50
	r = 0.7	89.35	77.77	34.37
	r = 0.9	**90.64**	**87.85**	**76.10**
CIFAR-10	r = 0.1	82.62	64.20	48.99
	r = 0.3	**82.97**	**77.68**	53.09
	r = 0.5	82.06	76.95	62.31
	r = 0.7	82.19	77.24	**62.92**
	r = 0.9	81.31	75.99	62.38

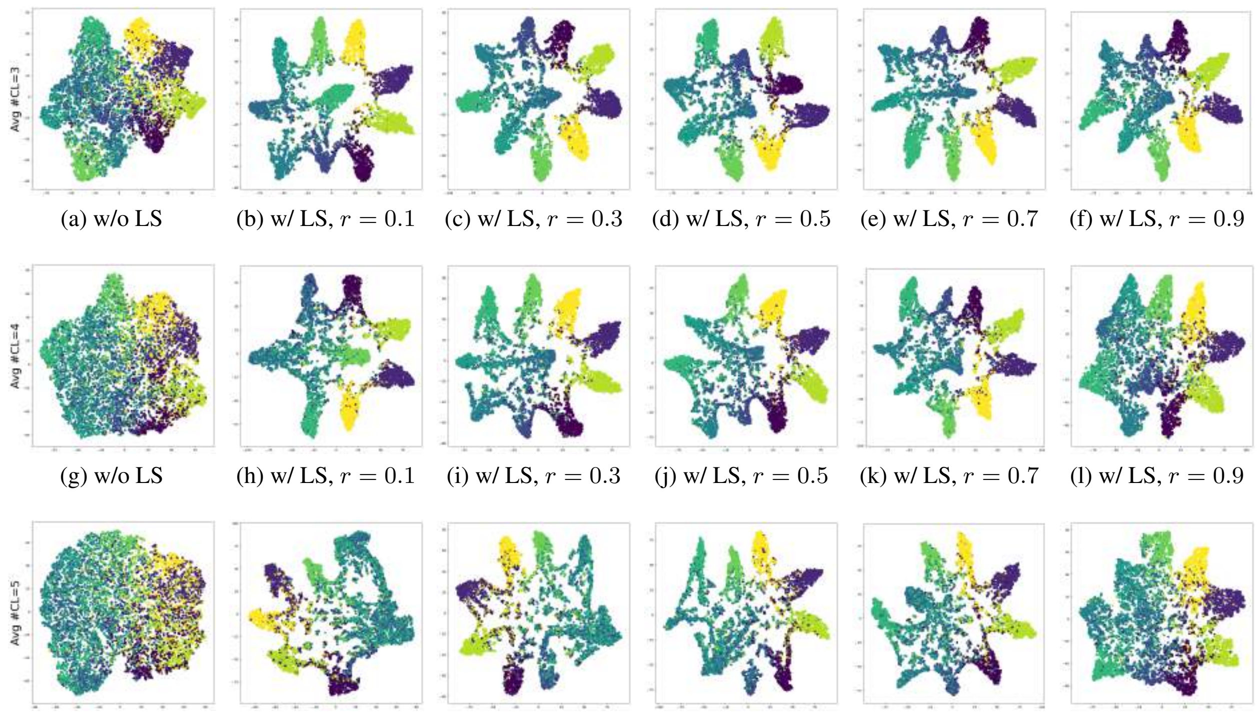
Table 4. Test accuracy comparisons of LS-PLL w.r.t. varying smoothing rate r on CIFAR-100 dataset at different noise levels (i.e., Avg.#CL = 7, 9, 11). The best results are highlighted in bold.

Dataset	SmoothRate	Avg.#CL=7	Avg.#CL=9	Avg.#CL=11
CIFAR-100	r = 0.1	37.62	27.40	16.40
	r = 0.3	38.00	20.00	22.70
	r = 0.5	43.50	25.80	22.10
	r = 0.7	48.67	38.80	25.10
	r = 0.9	**57.90**	**47.50**	**33.10**

method from Müller et al. (2019) to compare the pre-logits of basic neural architectures trained without label smoothing (i.e., w/o LS) and with label smoothing (i.e., w/ LS) on PLL datasets under various noise levels (i.e., Avg. #CL) in Figs. 1-4. We show Fig. 1 of CIFAR-10 in main paper, while the rest figures can be found in Appendix. The first column represents pre-logits of examples trained w/o LS, while the second to sixth column represent pre-logits of examples trained w/ LS under varying smoothing rate $r = 0.1, 0.3, 0.5, 0.7, 0.9$. Each row represents one kind of noise levels. From these figures, we can observe that the first column w/o LS shows broad clusters, while the last five columns w/ LS show tight clusters through various noise levels (i.e., Avg. #CL) on all PLL datasets. This is because label smoothing encourages activations of the penultimate layer to be close to the template of correct class and equally distant to the templates of incorrect classes on PLL datasets. Moreover, when noise levels increase (from the first row to the third row), clusters become more broad and fuzzy on all PLL datasets. In particular, the clusters w/o LS (the first column) become more broad and vague than that w/ LS (the last five columns). The results are consistent on all datasets across various neural architectures, which indicates that label smoothing is effective for the penultimate layer's representations of deep PLL regardless of architecture. Thus, it can be concluded that label smoothing is effective in learning distinguishable representations for deep PLL, and that the effect of label smoothing on pre-logits is independent of architectures, datasets and noise levels.

5.3. Experimental Results w.r.t. 'When'

We conduct experiments to validate when label smoothing helps deep PLL most by comparing the prediction accuracies of LS-PLL at varying smoothing rates in Table 3 and 4. From these results, we can observe that most datasets prefer high label smoothing rate. For example, on Fashion-MNIST, the best results under Avg. #CL = 3, 4, 5 are achieved at $r = 0.9, 0.9, 0.9$ respectively. Similar behavior can be observed on Kuzushiji-MNIST, and CIFAR-100. Meanwhile, some datasets seem to prefer middle to large label smoothing rates. For example, on CIFAR-10, the highest accuracies under Avg.#CL = 3, 4, 5 are achieved at $r = 0.3, 0.3, 0.7$ respectively. Moreover, the best results are consistent with the best effect of pre-logit representations in Fig. 2-4. For example, on Kuzushiji-MNIST, the best cluster effect under noise level Avg. #CL=3, 4, 5 is obtained at $r = 0.9, 0.9, 0.9$ respectively, while on CIFAR-10, the best cluster effect is



(a) w/o LS (b) w/ LS, $r = 0.1$ (c) w/ LS, $r = 0.3$ (d) w/ LS, $r = 0.5$ (e) w/ LS, $r = 0.7$ (f) w/ LS, $r = 0.9$

(g) w/o LS (h) w/ LS, $r = 0.1$ (i) w/ LS, $r = 0.3$ (j) w/ LS, $r = 0.5$ (k) w/ LS, $r = 0.7$ (l) w/ LS, $r = 0.9$

(m) w/o LS (n) w/ LS, $r = 0.1$ (o) w/ LS, $r = 0.3$ (p) w/ LS, $r = 0.5$ (q) w/ LS, $r = 0.7$ (r) w/ LS, $r = 0.9$

Figure 1. Visualization of pre-logits of *CIFAR-10*/ResNet-18 with various ambiguity degrees Avg.#CL=3 (first row), Avg.#CL=4 (second row), and Avg.#CL=5 (third row).

obtained at $r = 0.3, 0.3, 0.7$ respectively. Here, the best cluster effect means ten clusters are most clearly seperated with tight gathering. The results demonstrate that the best prediction accuracy can be achieved at a certain smoothing rate, which is supposed to be correlated with different datasets and various noise levels. The above experimental results corroborate our theoretical findings to some extent. For example, on CIFAR-10, given the definition of General Ambiguity Degree ϵ, we can figure out the value of ϵ under noise levels Avg. #CL=3, 4, 5 to be 0.0049, 0.0058, 0.0018 respectively. Given Eq. (8), we can further get the theoretical optimal smoothing rate $r_{opt} = 0.24, 0.26, 0.68$, which can be approximated by the empirical best smoothing rate $r = 0.3, 0.3, 0.7$ respectively. To sum up, these empirical results validate our theoretical findings, which in turn guides on the parameter choice of empirical smoothing rate.

6. Conclusion

This paper presents a novel insight into deep PLL from label smoothing. We study whether and when label smoothing helps deep PLL from both the theoretical and emprical aspects. Theoretically, we provide affirmative answers to these questions by proving the lower and upper bounds of the expected risk with respect to label smoothing on PLL, and deriving the optimal smoothing rate. Practically, we propose a benchmark solution, design a novel algorithm (i.e., LS-PLL), and conduct extensive experiments on benchmark datasets to validate that label smoothing does help deep PLL in improving performance and learning distinguishable representations regardless of achitectures, and the best results can be achieved when the empirical smoothing rate approximately approaches the optimal smoothing rate in theoretical findings. This research also bridges the gap of existing deep PLL methods for lacking theoretical guarantee on the classification performance and the label noise by forging a connection between the optimal smoothing rate and the generalized ambiguity degree, which sheds light on the parameter choice of smoothing rate in empirical studies.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Grants No.62072257 and the Australian Research Council Under Grants DP22010371, LE220100078.

Impact Statement

This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

References

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3:463–482, 2002.

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, M., Gong, X., Jin, Y., and Hu, W. Relation preference oriented high-order sampling for recommendation. In *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining, WSDM*, pp. 105–113, 2023.

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.

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

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

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.

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

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical Report 32–33, RAND Corporation, 2009.

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

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

Liu, W. and Tsang, I. W. Making decision trees feasible in ultrahigh feature and label dimensions. *The Journal of Machine Learning Research*, 18:81:1–81:36, 2017.

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. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 41(2):408–422, 2019.

Lukasik, M., Bhojanapalli, S., Menon, A. K., and Kumar, S. Does label smoothing mitigate label noise? In *ICML*, pp. 6448–6458, 2020.

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., 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.

Mao, Y., Wang, Z., Liu, W., Lin, X., and Hu, W. Task variance regularized multi-task learning. *IEEE Trans. Knowl. Data Eng.*, 35(8):8615–8629, 2023.

Maurer, A. A vector-contraction inequality for rademacher complexities. In *ALT*, volume 9925, pp. 3–17, 2016.

Mohri, M., Rostamizadeh, A., and Talwalkar, A. *Foundations of Machine Learning*. Adaptive computation and machine learning. MIT Press, 2012.

Müller, R., Kornblith, S., and Hinton, G. E. When does label smoothing help? In *NeurIPS*, pp. 4696–4705, 2019.

Robbins, Herbert, and Monro, S. A stochastic approximation method. *The Annals of Mathematical Statistics*, 22 (03):400–07, 1951.

Shalev-Shwartz, S. and Ben-David, S. *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *CVPR*, pp. 2818–2826, 2016.

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

Wang, H., Xiao, R., Li, Y., Feng, L., Niu, G., Chen, G., and Zhao, J. Pico: Contrastive label disambiguation for partial label learning. In *ICLR*, 2022.

Warnke, L. On the method of typical bounded differences. *Combinatorics Probability and Computing*, 25(2):269–299, 2016.

Wei, J., Liu, H., Liu, T., Niu, G., Sugiyama, M., and Liu, Y. To smooth or not? when label smoothing meets noisy labels. In *ICML*, pp. 23589–23614, 2022.

Wen, H., Cui, J., Hang, H., Liu, J., Wang, Y., and Lin, Z. Leveraged weighted loss for partial label learning. In *ICML*, volume 139, pp. 11091–11100. PMLR, 2021.

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

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

Xu, N., Qiao, C., Geng, X., and Zhang, M. Instance-dependent partial label learning. In *NeurIPS*, pp. 27119–27130, 2021.

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

Yan, Y. and Guo, Y. Mutual partial label learning with competitive label noise. In *ICLR*, 2023.

Yao, Y., Gong, C., Deng, J., and Yang, J. Network cooperation with progressive disambiguation for partial label learning. In *ECML*, volume 12458, pp. 471–488, 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 *SIGKDD*, 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. Definitons and Proofs

Definition A.1 (Rademacher complexity)**.** Let $\mathcal{F} = \{f : \mathcal{X} \to \mathbb{R}\}$ denote a class of measurable functions, i.e., hypothesis space, then the Rademacher complexity w.r.t. $\mathcal{F}$, which quantifies how much $f \in \mathcal{F}$ is correlated with a noise sequence of length n, can be defined as follows:

$$\mathfrak{R}_n(\mathcal{F}) = \mathbb{E}_{\hat{\mathcal{S}}}\left[\hat{\mathfrak{R}}_n(\mathcal{F})\right]$$

where $\hat{\mathfrak{R}}_n(\mathcal{F}) = \mathbb{E}_\sigma\left[\sup_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^n \sigma_i f(\mathbf{x}_i)|\hat{\mathcal{S}}\right]$, σ_i denotes n independent uniform $\{\pm 1\}$-valued random variables. $\hat{\mathcal{S}} = \{\mathbf{x}_i\}_{i=1}^n$ are independent samples.

Definition A.2 (ρ-Lipschitz)**.** Let $f : \mathbb{R}^m \to \mathbb{R}$ be a function. If

$$\left|f(\mathbf{t}) - f(\mathbf{t}^{'})\right| \le \rho\left\|(t_1 - t_1^{'}, \dots, t_m - t_m^{'})\right\|, \forall \mathbf{t}, \mathbf{t}^{'} \in \mathbb{R}^m,$$

function f is ρ-Lipschitz w.r.t. a norm $\|\cdot\|$ in $\mathbb{R}^m$. The ℓ_p-norm of a vector $\mathbf{t} = (t_1, \dots, t_m)$ is defined as $\left\|\mathbf{t}\right\|_p = \left[\sum_{j=1}^m |t_j|^p\right]^{\frac{1}{p}}$.

Proof of Theorem 3.3

Proof.

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\ell\big(\mathbf{f}(\mathbf{X}), \mathbf{Y}^{LS}\big)\big]\\
=&\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\Big[\sum_{j\in\mathbf{Y}} \mathbf{Y}^{LS,j}\Big(-f^j + \log\sum_{k\in[L]} e^{f^k}\Big)\Big]\\
=&\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\Big[\sum_{j\in\mathbf{Y}} \big((1-r)\cdot\mathbb{I}_{j=\mathbf{Y}^*} + \frac{r}{|\mathbf{Y}|}\big)\\
&\qquad\qquad\cdot\big(-f^j + \log\sum_{k\in[L]} e^{f^k}\big)\Big]\\
=&\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\Big[(1-r)\sum_{j\in\mathbf{Y}}\mathbb{I}_{j=\mathbf{Y}^*}\big(-f^j + \log\sum_{k\in[L]} e^{f^k}\big)\\
&\quad + \frac{r}{|\mathbf{Y}|}\sum_{j\in\mathbf{Y}}\big(-f^j + \log\sum_{k\in[L]} e^{f^k}\big)\Big]\\
=&\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[(1-r)\ell\big(\mathbf{f}(\mathbf{X}), \mathbf{Y}^*\big)\big]\\
&\quad + \mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\frac{r}{|\mathbf{Y}|}\sum_{j\in\mathbf{Y}}\ell\big(\mathbf{f}(\mathbf{X}), j\big)\big]
\end{aligned}$$

□

Proof of Theorem 3.5

Proof. Following Theorem 3.3, we can further derive that

$$
\begin{aligned}
&\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^{LS}\big)\big]\\
&=\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[(1-r)\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^*\big)\big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\frac{r}{|\mathbf{Y}|}\sum_{j\in\mathbf{Y}}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\big]\\
&=\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[(1-r)\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^*\big)\big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*,\mathbf{Z})}\Big[\frac{r}{c+1}\Big(\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^*\big)+\sum_{j\in\mathbf{Z}}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\Big)\Big]\\
&=\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[(1-r+\frac{r}{c+1})\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^*\big)\big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*,\mathbf{Z})}\Big[\frac{r}{c+1}\sum_{j\in\mathbf{Z}}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\Big]\\
&=\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[(1-r+\frac{r}{c+1})\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^*\big)\big]\\
&\quad+\frac{r}{c+1}\int_x\sum_{y\in[L]}\sum_{Z\subset\mathcal{Y}\backslash y}P(x,y)P(Z|x,y)\sum_{j\in Z}\ell\big(\mathbf{f}(x),j\big)dx\\
&=\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[(1-r+\frac{r}{c+1})\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^*\big)\big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[\frac{r}{c+1}\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}P(Z|\mathbf{X},\mathbf{Y}^*)\sum_{j\in Z}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\Big]\\
&=\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[(1-r+\frac{r}{c+1})\frac{1}{1-r^*}\big[\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^{*LS}\big)\\
&\qquad\qquad-\frac{r^*}{L}\sum_{j\in[L]}\ell\big(\mathbf{f}(x),j\big)\big]\Big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[\frac{r}{c+1}\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}P(Z|\mathbf{X},\mathbf{Y}^*)\sum_{j\in Z}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\Big]\\
&=\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[(1-r+\frac{r}{c+1})\frac{1}{1-r^*}\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^{*LS}\big)\Big]\\
&\quad-\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[(1-r+\frac{r}{c+1})\frac{1}{1-r^*}\ \frac{r^*}{L}\sum_{j\in[L]}\ell\big(\mathbf{f}(x),j\big)\Big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[\frac{r}{c+1}\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}P(Z|\mathbf{X},\mathbf{Y}^*)\sum_{j\in Z}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\Big]\\
&=\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[(1-r+\frac{r}{c+1})\frac{1}{1-r^*}\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^{*LS}\big)\Big]\\
&\quad-\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[(1-r+\frac{r}{c+1})\frac{1}{1-r^*}\cdot\frac{r^*}{L}\cdot\\
&\qquad\frac{1}{\binom{L}{c}}\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}\big[\sum_{j\in Z}\ell\big(\mathbf{f}(\mathbf{X}),j\big)+\sum_{j\in\bar{Z}}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\big]\Big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[\frac{r}{c+1}\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}P(Z|\mathbf{X},\mathbf{Y}^*)\sum_{j\in Z}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\Big]
\end{aligned}
\tag{16}
$$

For simplicity, let $t_1=\Big(1-r+\frac{r}{c+1}\Big)\frac{1}{1-r^*}$, $t_2=\frac{t_1}{\binom{L}{c}}\cdot\frac{r^*}{L}$, $t_3=\Big(\frac{r\epsilon}{c+1}-t_2\Big)$, then Eq. (16) can be reformulated as follows:

$$
\begin{aligned}
&Eq.\ (16)\\
&=\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[t_1\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^{*LS}\big)\Big]\\
&\quad-\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[t_2\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}\big[\sum_{j\in Z}\ell\big(\mathbf{f}(\mathbf{X}),j\big)+\sum_{j\in\bar{Z}}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\big]\Big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[\frac{r}{c+1}\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}P(Z|\mathbf{X},\mathbf{Y}^*)\sum_{j\in Z}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\Big]\\
&=\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[t_1\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^{*LS}\big)\Big]\\
&\quad-\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[t_2\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}\big[\sum_{j\in Z}\ell\big(\mathbf{f}(\mathbf{X}),j\big)+\sum_{j\in\bar{Z}}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\big]\Big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[\frac{r}{c+1}\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}P(Z|\mathbf{X},\mathbf{Y}^*)\sum_{j\in Z}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\Big]\\
&\leq\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[t_1\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^{*LS}\big)\Big]\\
&\quad-\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[t_2\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}\big[\sum_{j\in Z}\ell\big(\mathbf{f}(\mathbf{X}),j\big)+\sum_{j\in\bar{Z}}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\big]\Big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[\frac{r}{c+1}\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}\epsilon\sum_{j\in Z}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\Big]\\
&=\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[t_1\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^{*LS}\big)\Big]\\
&\quad-\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[t_2\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}\sum_{j\in\bar{Z}}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\Big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[t_3\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}\sum_{j\in Z}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\Big]
\end{aligned}
\tag{17}
$$

By combining Eq. (16) and Eq. (17), we have,

$$
\begin{aligned}
&\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^{LS}\big)\big]\\
&\leq\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[t_1\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^{*LS}\big)\Big]\\
&\quad-\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[t_2\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}\sum_{j\in\bar{Z}}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\Big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\Big[t_3\sum_{Z\subset\mathcal{Y}\backslash\mathbf{Y}^*}\sum_{j\in Z}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\Big]
\end{aligned}
\tag{18}
$$

With a little bit math, we can derive Eq. (7) in Theorem 3.5. □

Proof of Theorem 3.7

Proof. Following Theorem 3.3, we can derive that

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^{LS}\big)\big]-\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^{*LS}\big)\big]\\
&=\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[(1-r)\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^*\big)\big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\frac{r}{|\mathbf{Y}|}\sum_{j\in\mathbf{Y}}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\big]\\
&\quad-\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[(1-r^*)\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^*\big)\big]\\
&\quad-\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\frac{r^*}{L}\sum_{j\in[L]}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\big]\\
&=\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[(r^*-r)\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^*\big)\big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[(\frac{r}{|\mathbf{Y}|}-\frac{r^*}{L})\sum_{j\in\mathbf{Y}}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\big]\\
&\quad-\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\frac{r^*}{L}\sum_{j\in\bar{\mathbf{Y}}}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\big]
\end{aligned}\tag{19}$$

From Eq. (19) with a little bit math, we can get

$$\begin{aligned}
&\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^{LS}\big)\big]-\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^{*LS}\big)\big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[\frac{r^*}{L}\sum_{j\in\bar{\mathbf{Y}}}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\big]\\
&=\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[(r^*-r)\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^*\big)\big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\big[(\frac{r}{c+1}-\frac{r^*}{L})\sum_{j\in\mathbf{Y}}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\big]\\
&=\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\big[\underbrace{(r^*-r+\frac{r}{c+1}-\frac{r^*}{L})}_{\text{c1}}\ell\big(\mathbf{f}(\mathbf{X}),\mathbf{Y}^*\big)\big]\\
&\quad+\mathbb{E}_{(\mathbf{X},\mathbf{Z})}\big[\underbrace{(\frac{r}{c+1}-\frac{r^*}{L})}_{c2}\sum_{j\in\mathbf{Z}}\ell\big(\mathbf{f}(\mathbf{X}),j\big)\big]
\end{aligned}\tag{20}$$

Next, we prove $c2>0$. Given $r=\frac{(c+1)r^*}{\Delta}$ and $\Delta=L\binom{L}{c}\epsilon(1-r^*)+cr^*$, we can get $\frac{r}{c+1}=\frac{r^*}{\Delta}$. Then, given $r*<\frac{L-L\binom{L}{c}\epsilon}{c-L\binom{L}{c}\epsilon}$ as a result of $r^*\in[0,1]$ and $\frac{L-L\binom{L}{c}\epsilon}{c-L\binom{L}{c}\epsilon}>1$, we can get $\Delta<L$. Thus, we can get $\frac{r}{c+1}>\frac{r^*}{L}$, which verifies that $c2>0$.

To get the upper bound, it is required that $c1>0$, by derivation, we can get a condition for the upperbound to be satisfied in Theorem 3.7, which is smoothing rate $\epsilon>\delta$ where $\delta=\frac{Lc-(L-1)cr^*}{L(L-1)\binom{L}{c}(1-r^*)}$. Combining with $\epsilon\in(0,1)$, we can get the condition in Theorem 3.7, i.e., $\epsilon\in(\delta,1)$ and $\delta=\frac{Lc-(L-1)cr^*}{L(L-1)\binom{L}{c}(1-r^*)}$.

By applying $c1>0, c2>0$ in Eq. (20) with a little bit math, we can derive Eq. (10) in Theorem 3.7. □

Proof of Lemma 3.8

Proof. If $\ell\big(\mathbf{f}(x),y\big)$ is ρ-Lipschitz for every y, then for any $\mathbf{f}_1(x),\mathbf{f}_2(x)$, we have $|\ell\big(\mathbf{f}_1(x),y\big)-\ell\big(\mathbf{f}_2(x),y\big)\leq\rho|\mathbf{f}_1(x)-\mathbf{f}_2(x)|$.

Thereby, we can get,

$$\begin{aligned}
&|\ell\big(\mathbf{f}_1(x),\boldsymbol{Y}^{LS}\big)-\ell\big(\mathbf{f}_2(x),\boldsymbol{Y}^{LS}\big)|\\
&=|(1-r)\ell\big(\mathbf{f}_1(x),y\big)+\frac{r}{|Y|}\sum_{j\in Y}\ell\big(\mathbf{f}_1(x),j\big)\\
&\quad-(1-r)\ell\big(\mathbf{f}_2(x),y\big)-\frac{r}{|Y|}\sum_{j\in Y}\ell\big(\mathbf{f}_2(x),j\big)\\
&=\big|(1-r)\big[\ell\big(\mathbf{f}_1(x),y\big)-\ell\big(\mathbf{f}_2(x),y\big)\big]\\
&\quad+\frac{r}{|Y|}\sum_{j\in Y}\big[\ell\big(\mathbf{f}_1(x),j\big)-\ell\big(\mathbf{f}_2(x),j\big)\big]\big|\\
&\leq(1-r)\big|\ell\big(\mathbf{f}_1(x),y\big)-\ell\big(\mathbf{f}_2(x),y\big)\big|\\
&\quad+\frac{r}{|Y|}\sum_{j\in Y}\big|\ell\big(\mathbf{f}_1(x),j\big)-\ell\big(\mathbf{f}_2(x),j\big)\big|\\
&\leq\rho|\mathbf{f}_1(x)-\mathbf{f}_2(x)|
\end{aligned}$$

Thus, we have $|\ell\big(\mathbf{f}_1(x),\boldsymbol{Y}^{LS}\big)-\ell\big(\mathbf{f}_2(x),\boldsymbol{Y}^{LS}\big)|\leq\rho|\mathbf{f}_1(x)-\mathbf{f}_2(x)|$, which concludes the proof. □

Proof of Lemma 3.9

Proof. Given Definition A.1, we have $\mathfrak{R}_n(\mathcal{G})=\mathbb{E}_{(\mathbf{X},\mathbf{Y})}\mathbb{E}_\sigma\left[\sup\limits_{g\in\mathcal{G}}\frac{1}{n}\sum_{i=1}^n\sigma_i g(x_i,Y_i)\right]$. By defining $\ell\circ\mathcal{F}\doteq\{\ell\circ\mathbf{f}|\mathbf{f}\in\mathcal{F}\}\doteq\{(x,y)\mapsto\ell(\mathbf{f}(x),y)|\mathbf{f}\in\mathcal{F}\}$, we have $\mathfrak{R}_n(\ell\circ\mathcal{F})=\mathbb{E}_{(\mathbf{X},\mathbf{Y}^*)}\mathbb{E}_\sigma\left[\sup\limits_{\mathbf{f}\in\mathcal{F}}\frac{1}{n}\sum_{i=1}^n\sigma_i\ell(\mathbf{f}(x_i),y_i)\right]$. Given $\sum_{j\in Y}Y^{LS,j}=1$ and $Y^{LS,j}\in[0,1]$, we have $\mathfrak{R}_n(\mathcal{G})\leq\mathfrak{R}_n(\ell\circ\mathcal{F})$. Since $\mathcal{F}_y\doteq\{\mathbf{f}:x\mapsto\mathbf{f}(x)|\mathbf{f}\in\mathcal{F},y\in Y\}$ and $\ell\big(\mathbf{f}(x),\mathbf{Y}^{LS}\big)$ is ρ-Lipschitz given Lemma 3.8 with respect to $\mathbf{f}(x)$ for all $y\in Y$, by the Rademacher vector contraction inequality (Maurer, 2016), we have $\mathfrak{R}_n(\ell\circ\mathcal{F})\leq\sqrt{2}\rho\sum_{y\in Y}\mathfrak{R}_n(\mathcal{F}_y)$. □

Proof of Theorem 3.10

Proof. In order to prove this theorem, we first show that the one direction $\sup_{f\in\mathcal{F}}\mathcal{R}_{exp}(f)-\mathcal{R}_{emp}(f)$ is bounded with probability of at least $1-\eta/2$, and the other direction can be verified in the same way. Suppose an example (x_i,Y_i) is replaced by another arbitrary example (x_i',Y_i'), then the change of $\sup_{\mathbf{f}\in\mathcal{F}}\mathcal{R}_{exp}(f)-\mathcal{R}_{emp}(f)$ is no greater than $M/2n$, since $\ell(\mathbf{f}(x),Y^{LS})$ is upper-bounded by M. By applying McDiarmid's inequality (Warnke, 2016), for any

$\eta > 0$, with probability of at least $1 - \eta/2$, we have,

$$\begin{aligned}&\sup_{f\in\mathcal{F}} \mathcal{R}_{exp}(f) - \mathcal{R}_{emp}(f)\\ &\leq \mathbb{E}\Big[\sup_{f\in\mathcal{F}} \mathcal{R}_{exp}(f) - \mathcal{R}_{emp}(f)\Big] + \frac{M}{2}\sqrt{\frac{\log\frac{2}{\eta}}{2n}}\end{aligned}$$

By applying symmetrization (Mohri et al., 2012), we have,

$$\mathbb{E}\Big[\sup_{f\in\mathcal{F}} \mathcal{R}_{exp}(f) - \mathcal{R}_{emp}(f)\Big] \leq 2\mathfrak{R}_n(\mathcal{G})$$

By further taking into account the other side $\sup_{f\in\mathcal{F}} \mathcal{R}_{emp}(f) - \mathcal{R}_{exp}(f)$, we have, for $\eta > 0$, with probability of at least $1 - \eta$, we have,

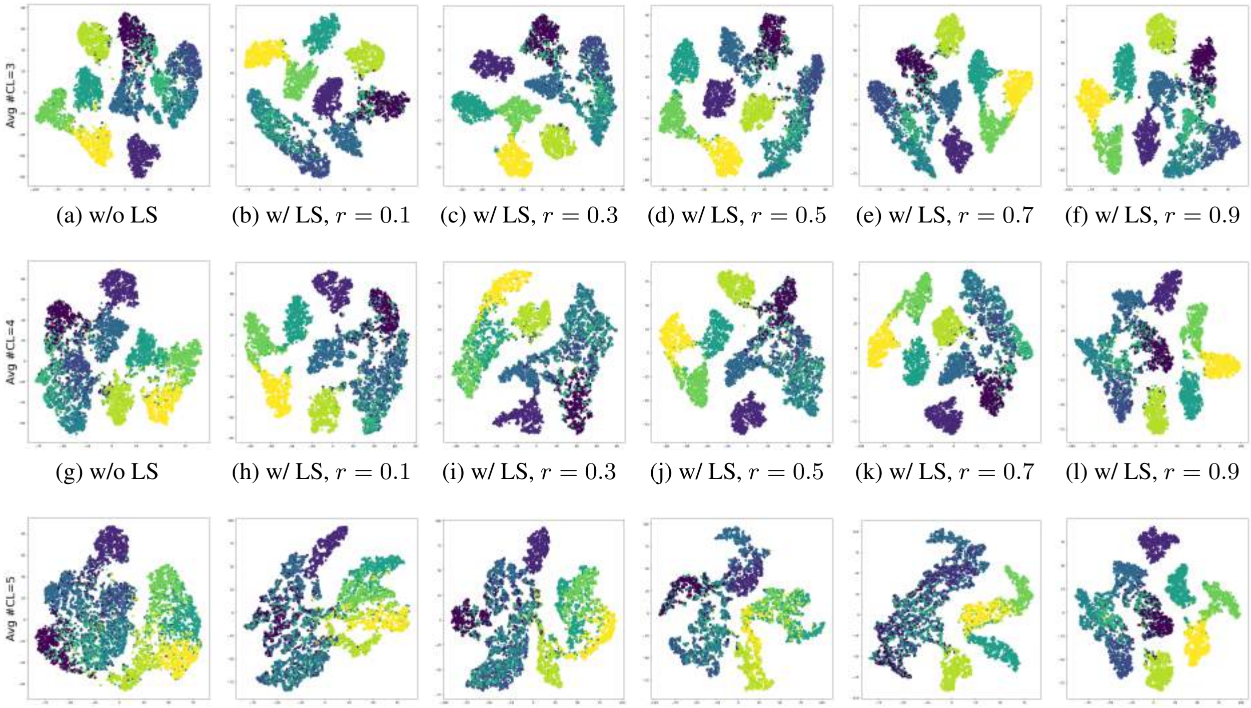
$$\sup_{f\in\mathcal{F}} \big|\mathcal{R}_{exp}(f) - \mathcal{R}_{emp}(f)\big| \leq 2\mathfrak{R}_n(\mathcal{G}) + \frac{M}{2}\sqrt{\frac{\log\frac{2}{\eta}}{2n}}$$

By combining Lemma 3.9, we conclude the proof. □

B. Tables and Figures

Table 5. Statistics of generated PLL datasets with competitive label noise.

Dataset	#Instances (Train)	#Instances (Test)	#Features	#Classes	Avg.#CL	Architecture
Fashion-MNIST	60,000	10,000	784	10	3, 4, 5	LeNet-5
Kuzushiji-MNIST	60,000	10,000	784	10	3, 4, 5	LeNet-5
CIFAR-10	50,000	10,000	3,072	10	3, 4, 5	ResNet-18
CIFAR-100	50,000	10,000	3,072	100	7, 9, 11	ResNet-56

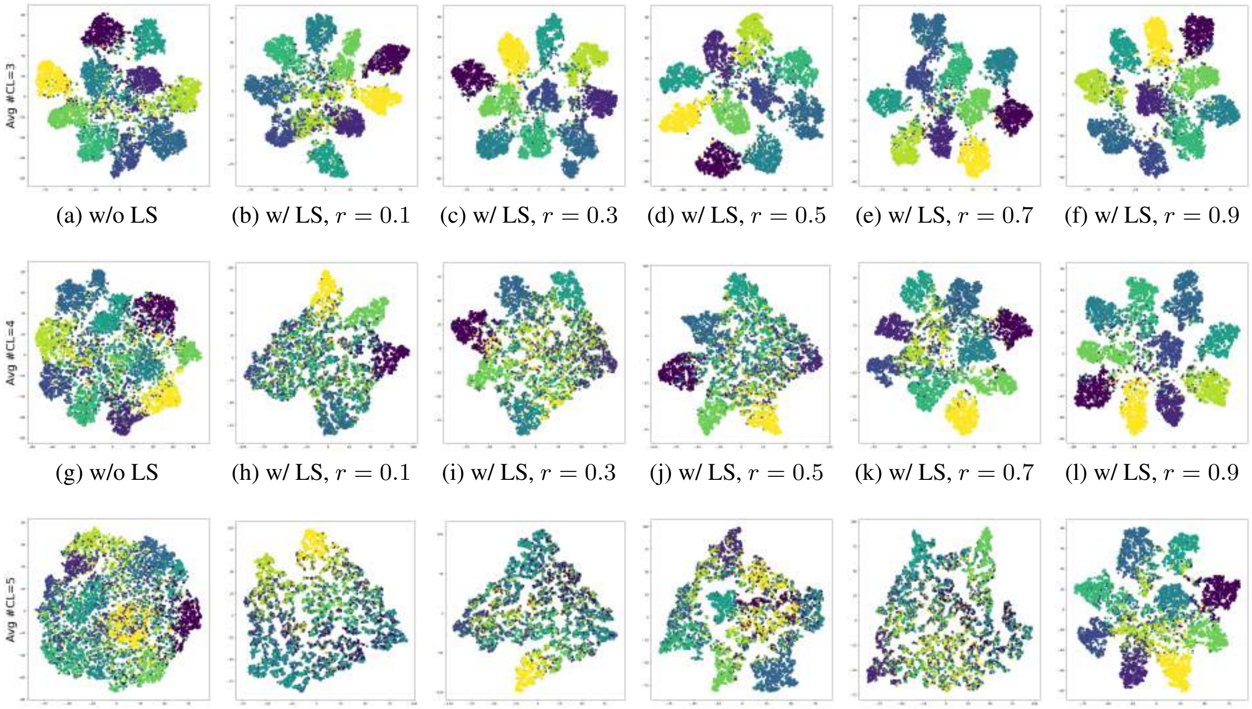


(a) w/o LS (b) w/ LS, $r = 0.1$ (c) w/ LS, $r = 0.3$ (d) w/ LS, $r = 0.5$ (e) w/ LS, $r = 0.7$ (f) w/ LS, $r = 0.9$

(g) w/o LS (h) w/ LS, $r = 0.1$ (i) w/ LS, $r = 0.3$ (j) w/ LS, $r = 0.5$ (k) w/ LS, $r = 0.7$ (l) w/ LS, $r = 0.9$

(m) w/o LS (n) w/ LS, $r = 0.1$ (o) w/ LS, $r = 0.3$ (p) w/ LS, $r = 0.5$ (q) w/ LS, $r = 0.7$ (r) w/ LS, $r = 0.9$

Figure 2. Visualization of pre-logits of *Fashion-MNIST*/LeNet-5 with various ambiguity degrees Avg.#CL=3 (first row), Avg.#CL=4 (second row), and Avg.#CL=5 (third row).

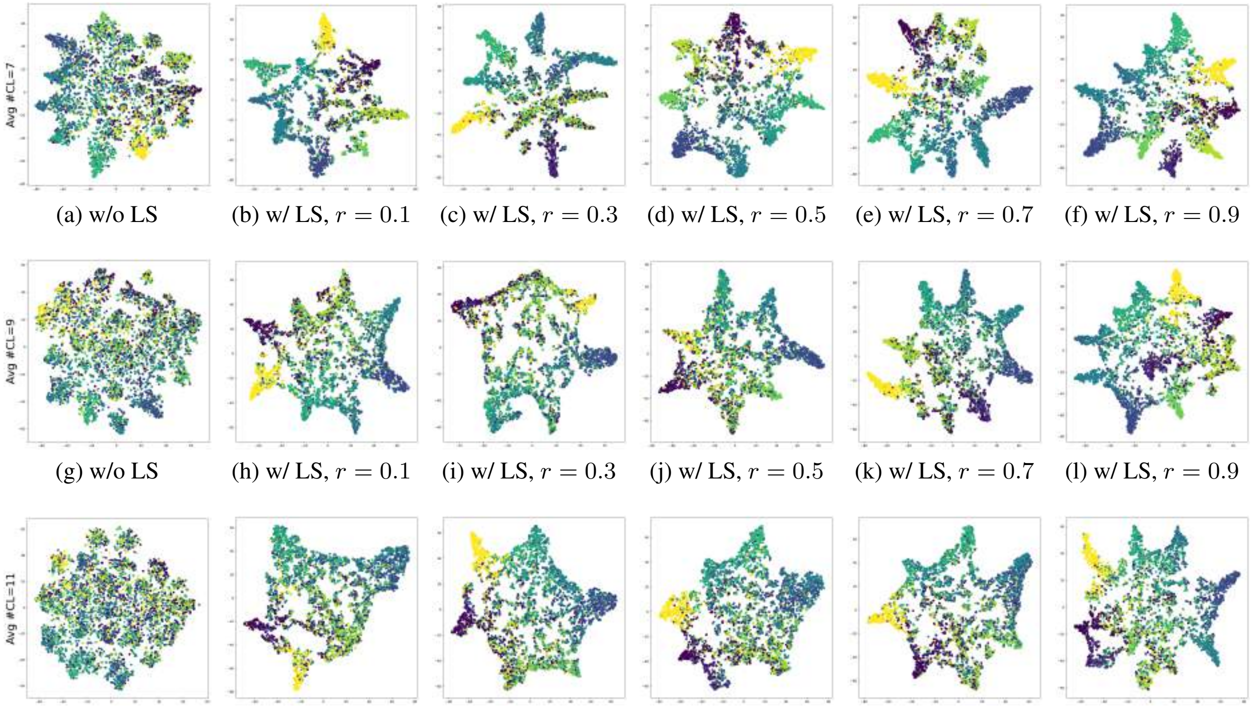


(a) w/o LS (b) w/ LS, $r = 0.1$ (c) w/ LS, $r = 0.3$ (d) w/ LS, $r = 0.5$ (e) w/ LS, $r = 0.7$ (f) w/ LS, $r = 0.9$

(g) w/o LS (h) w/ LS, $r = 0.1$ (i) w/ LS, $r = 0.3$ (j) w/ LS, $r = 0.5$ (k) w/ LS, $r = 0.7$ (l) w/ LS, $r = 0.9$

(m) w/o LS (n) w/ LS, $r = 0.1$ (o) w/ LS, $r = 0.3$ (p) w/ LS, $r = 0.5$ (q) w/ LS, $r = 0.7$ (r) w/ LS, $r = 0.9$

Figure 3. Visualization of pre-logits of *Kuzushiji-MNIST*/LeNet-5 with various ambiguity degrees Avg.#CL=3 (first row), Avg.#CL=4 (second row), and Avg.#CL=5 (third row).



(a) w/o LS (b) w/ LS, $r = 0.1$ (c) w/ LS, $r = 0.3$ (d) w/ LS, $r = 0.5$ (e) w/ LS, $r = 0.7$ (f) w/ LS, $r = 0.9$

(g) w/o LS (h) w/ LS, $r = 0.1$ (i) w/ LS, $r = 0.3$ (j) w/ LS, $r = 0.5$ (k) w/ LS, $r = 0.7$ (l) w/ LS, $r = 0.9$

(m) w/o LS (n) w/ LS, $r = 0.1$ (o) w/ LS, $r = 0.3$ (p) w/ LS, $r = 0.5$ (q) w/ LS, $r = 0.7$ (r) w/ LS, $r = 0.9$

Figure 4. Visualization of pre-logits of *CIFAR-100*/ResNet-56 with various ambiguity degrees Avg.#CL=7 (first row), Avg.#CL=9 (second row), and Avg.#CL=11 (third row).