Leveraged Weighted Loss For Partial Label Learning (Supplementary Material)

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

This file consists of supplementaries for both theoretical analysis and experiments. In Section A, we present the proof of Theorem 1 in Section 3. In Section B, we present more detailed settings of the numerical experiments including descriptions of datasets, compared methods, model architecture, and data generation procedures.

A. Proofs

We present all proofs for Section 3 here. For the sake of conciseness and readability, we denote $\vec{\mathcal{Y}}^y$ as the collection of all partial label sets containing the true label y, i.e. $\vec{\mathcal{Y}}^y := \{\vec{y} \in \vec{\mathcal{Y}} | y \in \vec{y}\}$.

In order to achieve the risk consistency result for the LW loss in Theorem 1, we first present in Theorem A.1 the risk consistency result for an arbitrary loss function $\bar{\mathcal{L}}(\vec{y}, g(x))$ under the generalized assumption that partial label sets follows the *label-specific* sampling.

Theorem A.1 *Denote $q_z := \mathrm{P}(z \in \vec{y} \,|\, Y = y, x)$. Then the partial loss function $\bar{\mathcal{L}}(\vec{y}, g(x))$ is risk-consistent with respect to the supervised loss function with the form*

$$\mathcal{L}(y, g(x)) = \sum_{\vec{y} \in \vec{\mathcal{Y}}^y} \prod_{s \in \vec{y}, s \neq y} q_s \prod_{t \notin \vec{y}} (1 - q_t) \bar{\mathcal{L}}(\vec{y}, g(x)), \quad (1)$$

where $\vec{\mathcal{Y}}^y := \{\vec{y} \in \vec{\mathcal{Y}} \,|\, y \in \vec{y}\}$ denotes the partial label set containing label y.

Proof 1 (of Theorem A.1) *For any $x \in \mathcal{X}$, there holds*

$$\begin{aligned}
&\bar{\mathcal{R}}(\bar{\mathcal{L}}, g(X)) \\
&= \mathbb{E}_{\vec{Y}|X}[\bar{\mathcal{L}}(\vec{Y}, g(x))|X = x] \\
&= \sum_{\vec{y} \in 2^{[K]}} \bar{\mathcal{L}}(\vec{y}, g(x)) \mathrm{P}(\vec{Y} = \vec{y} | X = x) \\
&= \sum_{\vec{y} \in 2^{[K]}} \bar{\mathcal{L}}(\vec{y}, g(x)) \sum_{y \in \boldsymbol{y}} \mathrm{P}(\vec{Y} = \vec{y}, Y = y | X = x) \\
&= \sum_{\vec{y} \in 2^{[K]}} \bar{\mathcal{L}}(\vec{y}, g(x)) \\
&\quad \cdot \sum_{y \in \boldsymbol{y}} \mathrm{P}(\vec{Y} = \vec{y} | Y = y, X = x) \mathrm{P}(Y = y | X = x) \\
&= \sum_{y=1}^{K} \mathrm{P}(Y = y | X = x) \\
&\quad \cdot \sum_{\vec{y} \in 2^{[K]}} \mathrm{P}(\vec{Y} = \vec{y} | Y = y, X = x) \bar{\mathcal{L}}(\vec{y}, g(x)),
\end{aligned}$$

and

$$\begin{aligned}
\mathcal{R}(\mathcal{L}, g(X)) &= \mathbb{E}_{Y|X}[\mathcal{L}(Y, g(x))|X = x] \\
&= \sum_{y=1}^{K} \mathcal{L}(y, g(x)) \mathrm{P}(Y = y | X = x).
\end{aligned}$$

Since $\mathrm{P}(\vec{Y} = \vec{y} | Y = y, X = x) = 0$ for $\vec{y}$ not containing y, if we have

$$\begin{aligned}
\mathcal{L}(y, g(x)) &= \sum_{\vec{y} \in 2^{[K]}} \mathrm{P}(\vec{Y} = \vec{y} | Y = y, X = x) \bar{\mathcal{L}}(\vec{y}, g(x)) \\
&= \sum_{\vec{y} \in \vec{\mathcal{Y}}^y} \mathrm{P}(\vec{Y} = \vec{y} | Y = y, X = x) \bar{\mathcal{L}}(\vec{y}, g(x)), \\
&= \sum_{\vec{y} \in \vec{\mathcal{Y}}^y} \prod_{s \in \vec{y}, s \neq y} q_s \prod_{t \notin \vec{y}} (1 - q_t) \bar{\mathcal{L}}(\vec{y}, g(x)),
\end{aligned}$$

then there holds

$$\bar{\mathcal{R}}(\bar{\mathcal{L}}, g(X)) = \mathcal{R}(\mathcal{L}, g(X)).$$

Besides, to prove Theorem 1, we need the following result shown in Lemma A.1.

Lemma A.1 *Let y be the true label of input x, $q_z := \mathrm{P}(z \in \vec{y} | Y = y, X = x)$ for $z \in \mathcal{Y}$, and $\vec{\mathcal{Y}}^y := \{\vec{y} \in \vec{\mathcal{Y}} | y \in \vec{y}\}$. Then there holds*

$$\sum_{\vec{y} \in \vec{\mathcal{Y}}^y} \prod_{s \in \vec{y}, s \neq y} q_s \prod_{t \notin \vec{y}} (1 - q_t) = 1.$$

[*]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.

Proof 2 (of Lemma A.1) *Since $q_y = 1$, we have*

$$\begin{aligned}
&\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\notin\vec{y}}(1-q_t)\\
&=\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y},s\neq y} 1\cdot q_s\prod_{t\notin\vec{y}}(1-q_t)\\
&=\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y},s\neq y} q_y\cdot q_s\prod_{t\notin\vec{y}}(1-q_t)\\
&=\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y}} q_s\prod_{t\notin\vec{y}}(1-q_t)\\
&=\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\mathrm{P}(\vec{Y}=\vec{y}|Y=y,X=x)\\
&=\sum_{\vec{y}\in\vec{\mathcal{Y}}}\mathrm{P}(\vec{Y}=\vec{y}|Y=y,X=x)\\
&=1,
\end{aligned}$$

where the second last equation holds since $\mathrm{P}(\vec{Y}=\vec{y}|Y=y,X=x)=0$ for $\vec{y}\notin\vec{\mathcal{Y}}^y$.

Proof 3 (of Theorem 1) *According to Theorem A.1, we have the partial loss function $\bar{\mathcal{L}}_\psi$ consistent with*

$$\begin{aligned}
&\mathcal{L}_\psi(y,g(x))\\
&=\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\notin\vec{y}}(1-q_t)\bar{\mathcal{L}}_\psi(\vec{y},g(x))\\
&=\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\notin\vec{y}}(1-q_t)\sum_{z\in\vec{y}} w_z\psi(g_z(x))\\
&+\beta\cdot\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\notin\vec{y}}(1-q_t)\sum_{z\notin\vec{y}} w_z\psi(-g_z(x)). \quad (2)
\end{aligned}$$

The first term on the right hand side of (2) *is*

$$\begin{aligned}
&\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\notin\vec{y}}(1-q_t)\sum_{z\in\vec{y}} w_z\psi(g_z(x))\\
&=\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\notin\vec{y}}(1-q_t)w_y\psi(g_y(x))\\
&+\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\notin\vec{y}}(1-q_t)\sum_{z\in\vec{y}\setminus\{y\}} w_z\psi(g_z(x))\\
&=\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\notin\vec{y}}(1-q_t)w_y\psi(g_y(x))\\
&+\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\sum_{z\in\vec{y}\setminus\{y\}}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\in[K]\setminus\vec{y}}(1-q_t)w_z\psi(g_z(x)).
\end{aligned} \quad (3)$$

By Lemma A.1, we have

$$\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\notin\vec{y}}(1-q_t)=1,$$

and therefore the first term in (3) *becomes*

$$\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\notin\vec{y}}(1-q_t)w_y\psi(g_y(x))=w_y\psi(g_y(x)). \quad (4)$$

For the second term in (3)*, since $z\neq y$ and $z\in\vec{y}$, we switch the summations, and achieve*

$$\begin{aligned}
&\sum_{\vec{y}\in 2^{[K]}}\sum_{z\in\vec{y}\setminus\{y\}}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\in[K]\setminus\vec{y}}(1-q_t)w_z\psi(g_z(x))\\
&=\sum_{z\neq y}\sum_{\vec{y}\in\vec{\mathcal{Y}}^z\cap\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\in[K]\setminus\vec{y}}(1-q_t)w_z\psi(g_z(x))\\
&=\sum_{z\neq y} w_z\psi(g_z(x))\sum_{\vec{y}\in\vec{\mathcal{Y}}^z\setminus\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y}} q_s\prod_{t\in[K]\setminus\vec{y}\setminus\{y\}}(1-q_t).
\end{aligned}$$

Without loss of generality, we assume $y=K$ for notational simplicity, and write

$$\begin{aligned}
&\sum_{z\neq y} w_z\psi(g_z(x))\sum_{\vec{y}\in\vec{\mathcal{Y}}^z\setminus\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y}} q_s\prod_{t\in[K]\setminus\vec{y}\setminus\{y\}}(1-q_t)\\
&=\sum_{z\in[K-1]} w_z\psi(g_z(x))\sum_{\vec{y}\in(2^{[K-1]})^z}\prod_{s\in\vec{y}} q_s\prod_{t\in[K-1]\setminus\vec{y}}(1-q_t)\\
&=\sum_{z\in[K-1]} w_z\psi(g_z(x))q_z\\
&\cdot\sum_{\vec{y}\in(2^{[K-1]})^z}\prod_{s\in\vec{y},s\neq z} q_s\prod_{t\in[K-1]\setminus\vec{y}}(1-q_t).
\end{aligned}$$

Applying Lemma A.1 with $\vec{\mathcal{Y}}=2^{[K-1]}$, we have

$$\sum_{\vec{y}\in(2^{[K-1]})^z}\prod_{s\in\vec{y},s\neq z} q_s\prod_{t\in[K-1]\setminus\vec{y}}(1-q_t)=1, \quad (5)$$

and therefore the second term in (3) *becomes*

$$\begin{aligned}
&\sum_{\vec{y}\in 2^{[K]}}\sum_{z\in\vec{y}\setminus\{y\}}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\in[K]\setminus\vec{y}}(1-q_t)w_z\psi(g_z(x))\\
&=\sum_{z\neq y} q_z w_z\psi(g_z(x)).
\end{aligned} \quad (6)$$

Similarly, by switching the summations, the second term on the right hand side of (2) *becomes*

$$\begin{aligned}
&\beta\cdot\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\notin\vec{y}}(1-q_t)\sum_{z\notin\vec{y}} w_z\psi(-g_z(x))\\
&=\beta\cdot\sum_{\vec{y}\in\vec{\mathcal{Y}}^y}\sum_{z\notin\vec{y}}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\notin\vec{y}}(1-q_t)w_z\psi(-g_z(x))\\
&=\beta\cdot\sum_{z\neq y}\sum_{\vec{y}\in\vec{\mathcal{Y}}^y\cap\vec{\mathcal{Y}}^z}\prod_{s\in\vec{y},s\neq y} q_s\prod_{t\notin\vec{y}}(1-q_t)w_z\psi(-g_z(x))
\end{aligned}$$

$$= \beta \cdot \sum_{z\neq y} \sum_{\vec{y}\in\vec{\mathcal{Y}}^z\setminus\vec{\mathcal{Y}}^y} \prod_{s\in\vec{y}} q_s \prod_{t\notin\vec{y},t\neq y} (1-q_t) w_z \psi(-g_z(x))$$

$$= \beta \cdot \sum_{z\neq y} \sum_{\vec{y}\in(2^{[K-1]})^z} \prod_{s\in\vec{y}} q_s \prod_{t\notin\vec{y}} (1-q_t) w_z \psi(-g_z(x))$$

$$= \beta \cdot \sum_{z\neq y} q_z \sum_{\vec{y}\in(2^{[K-1]})^z} \prod_{s\in\vec{y},s\neq z} q_s \prod_{t\notin\vec{y}} (1-q_t) w_z \psi(-g_z(x))$$

$$= \beta \cdot \sum_{z\neq y} q_z w_z \psi(-g_z(x)), \tag{7}$$

where the last equality holds according to (5)*.*

By combining (4)*,* (6)*,* (7)*, we have*

$$\begin{aligned}\mathcal{L}_\psi(y, g(x)) &= w_y \psi(g_y(x)) + \sum_{z\neq y} q_z w_z \psi(g_z(x)) \\ &\quad + \beta \cdot \sum_{z\neq y} q_z w_z \psi(-g_z(x)) \\ &= w_y \psi(g_y(x)) \\ &\quad + \sum_{z\neq y} q_z w_z \big[\psi(g_z(x)) + \beta\psi(-g_z(x))\big].\end{aligned}$$

Before proving Theorem 2, we define the inner risk of loss function $\mathcal{L}_\psi$ by

$$\mathcal{C}_{\mathcal{L}_\psi}(g) := \mathbb{E}_{Y|X}\mathcal{L}_\psi(Y, g(X)) = \sum_{y\in[K]} p_y \mathcal{L}_\psi(Y, g(X)),$$

where $p_y := \mathrm{P}(Y = y\,|x)$.

Proof 4 (of Theorem 2) *By Theorem 1, we can write the inner risk induced by the supervised loss $\mathcal{L}_\psi$ as*

$$\begin{aligned}\mathcal{C}_{\mathcal{L}_\psi}(g) &:= \mathbb{E}_{Y|X}\mathcal{L}_\psi(Y, g(X)) \\ &= \sum_{y\in[K]} p_y \Big(w_y q_y \psi(g_y(x)) \\ &\quad + \sum_{z\neq y} w_z q_z \big[\psi(g_z(x)) + \beta\psi(-g_z(x))\big] \Big) \\ &= \sum_{y\in[K]} p_y w_y q_y \psi(g_y(x)) \\ &\quad + \sum_{y\in[K]} p_y \sum_{z\neq y} w_z q_z \big[\psi(g_z(x)) + \beta\psi(-g_z(x))\big] \\ &= \sum_{y\in[K]} p_y w_y q_y \psi(g_y(x)) \\ &\quad + \sum_{z\in[K]} \sum_{y\neq z} p_y w_z q_z \big[\psi(g_z(x)) + \beta\psi(-g_z(x))\big] \\ &= \sum_{y\in[K]} p_y w_y q_y \psi(g_y(x)) \\ &\quad + \sum_{y\in[K]} \sum_{z\neq y} p_z w_y q_y \big[\psi(g_y(x)) + \beta\psi(-g_y(x))\big] \\ &= \sum_{y\in[K]} \Big(p_y w_y q_y \psi(g_y(x)) \\ &\quad + (1-p_y) w_y q_y \big[\psi(g_y(x)) + \beta\psi(-g_y(x))\big] \Big),\end{aligned}$$

where $p_y := \mathrm{P}(Y = y\,|\,x)$.

Due to the symmetric property of $\psi(\cdot)$, we have

$$\begin{aligned}\mathcal{C}_{\mathcal{L}_\psi}(g) &= \sum_{y\in[K]} \Big(p_y w_y q_y \psi(g_y(x)) \\ &\quad + (1-p_y) w_y q_y \big[\beta + (1-\beta)\psi(g_y(x))\big] \Big) \\ &= \sum_{y\in[K]} w_y q_y (\beta p_y - (\beta-1))\psi(g_y) + C_1,\end{aligned}$$

where $C_1 := \sum_{y\in[K]} \beta(1-p_y) w_y q_y$.

Next, we consider the constraint comparison method (CCM) (Lee et al., 2004) defined by

$$\mathcal{L}_{CCM}(y, g(x)) := \sum_{k\neq y} \psi(-g_k(x))$$

with the constraint $\sum_{k\in[K]} g_k = 0$. The inner risk induced by the constraint comparison method (CCM) has the form

$$\begin{aligned}\mathcal{C}_{CCM}(g) &:= \mathbb{E}_{Y|X}\mathcal{L}_{CCM}(X, Y) \\ &= \sum_{y\in[K]} p_y \sum_{z\neq y} \psi(-g_z) \\ &= \sum_{y\in[K]} \sum_{z\neq y} p_y \psi(-g_z) \\ &= \sum_{z\in[K]} \sum_{y\neq z} p_y \psi(-g_z) \\ &= \sum_{y\in[K]} \sum_{z\neq y} p_z \psi(-g_y) \\ &= \sum_{y\in[K]} (1-p_y)\psi(-g_y).\end{aligned}$$

Since $\psi(\cdot)$ is symmetric, we have

$$\begin{aligned}\mathcal{C}_{CCM}(g) &= \sum_{y\in[K]} (1-p_y)(1-\psi(g_y)) \\ &= \sum_{y\in[K]} (p_y - 1)\psi(g_y) + C_2,\end{aligned}$$

where $C_2 := 1 - p_y$.

Denote $y^* := \max_{y\in[K]} p_y$. We have $p_{y^*} = 1$. By Section 3.4, we have $\arg\max_{y\in[K]} w_y = \arg\max_{y\in[K]} p_y$. Then when $\beta > 0$, there holds

$$\arg\max_{y\in[K]} w_y q_y (\beta p_y - (\beta-1)) = \arg\max_{y\in[K]} (p_y - 1).$$

which implies optimizing $\mathcal{L}_{LW}$ and $\mathcal{L}_{CCM}$ achieves the same classifier. According to Example 3 in Section 5.3 of (Tewari & Bartlett, 2007), when ψ is differentiable, $\mathcal{L}_{CCM}$ is proved to be consistent in the multi-class classification setting. Therefore, optimizing (12) *will also lead to the Bayes classifier, which implies when there holds*

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

there also holds

$$\mathcal{R}(\mathcal{L}_{0\text{-}1}, \hat{g}_n) \to \mathcal{R}^*, \tag{8}$$

where $\mathcal{L}_{0\text{-}1}$ is the multi-class supervised loss. This finishes the proof.

B. Supplementary for Experiments

B.1. Descriptions of Datasets

B.1.1. Benchmark Datasets

In Section 4.1.1, we use four widely-used benchmark datasets, i.e. MNIST(LeCun et al., 1998), Kuzushiji-MNIST (Clanuwat et al., 2018), Fashion-MNIST(Xiao et al., 2017), CIFAR-10(Krizhevsky et al., 2009). The characteristics of these datasets are reported in Table 1. We concisely describe these nine datasets as follows.

- MNIST: It is a 10-class dataset of handwritten digits, i.e. 0 to 9. Each data is a 28×28 grayscale image.
- Fashion-MNIST: It is also a 10-class dataset. Each instance is a fashion item from one of the 10 classes, which are T-shirt/top, trouser, pullover, dress, sandal, coat, shirt, sneaker, bag, and ankle boot. Moreover, each image is a 28×28 grayscale image.
- Kuzushiji-MNIST: Each instance is a 28×28 grayscale image associated with one label of 10-class cursive Japanese (Kuzushiji) characters.
- CIFAR-10: Each instance is a $32 \times 32 \times 3$ colored image in RGB format. It is a ten-class dataset of objects including airplane, bird, automobile, cat, deer, frog, dog, horse, ship, and truck.

B.1.2. Real Datasets

In Section 4.1.2, we use five real-world partially labeled datasets (Lost, BirdSong, MSRCv2, Soccer Player, Yahoo! News). Detailed descriptions are shown as follows.

- Lost, Soccer Player and Yahoo! News: They corp faces in images or video frames as instances, and the names appearing on the corresponding captions or subtitles are considered as candidate labels.
- MSRCv2: Each image segment is treated as a sample, and objects appearing in the same image are regarded as candidate labels.
- BirdSong: Birds' singing syllables are regarded as instances and bird species who are jointly singing during any ten seconds are represented as candidate labels.

Tabel 2 includes the average number of candidate labels (Avg. # CLs) per instance.

B.2. Compared Methods

The compared partial label methods are listed as follows.

IPAL (Zhang & Yu, 2015) : It is a non-parametric method that uses the label propagation strategy to iteratively update the confidence of each candidate label. The suggested configuration is as follows: the balancing coefficient $\alpha = 0.95$, the number of nearest neighbors considered $k = 10$, and the number of iterations $T = 100$.

PALOC ((Wu & Zhang, 2018)): It adapts the popular one-vs-one decomposition strategy to solve the partial label problem. The suggested configuration is the balancing coefficient $\mu = 10$ and the SVM model.

PLECOC ((Zhang et al., 2017)): It transforms the partial label learning problem to a binary label problem by E-COC coding matrix. The suggested configuration is codeword length $L = \lceil 10 \log 2(q) \rceil$ and SVM model. Moreover, the eligibility parameter τ is set to be one-tenth of the number of training instances (i.e. $\tau = |D|/10$).

Hyper-parameters for these three methods are selected through a 5-fold cross-validation.

Next, we list three compared partial label methods based on neural network models.

PRODEN((Lv et al., 2020)): It propose a novel estimator of the classification risk and a progressive identification algorithm for approximately minimizing the proposed risk estimator. The parameters is selected through grid search, where the learning rate $lr \in \{10^{-5}, 10^{-4}, \ldots, 10^{-1}\}$ and weight decay $wd \in \{10^{-6}, 10^{-4}, \ldots, 10^{-2}\}$. The optimizer is stochastic gradient descent (SGD) with momentum 0.9.

RC & CC((Feng et al., 2020)): The former method is a novel risk-consistent partial label learning method and the latter one is classifier-consistent based on the generation model. For the two methods, the suggested parameter grids of learning rate and weight decay are both $\{10^{-6}, 10^{-5}, \ldots, 10^{-1}\}$. They are implemented by PyTorch and the Adam optimizer.

For all these three compared methods, hyper-parameters are selected so as to maximize the accuracy on a validation set, constructed by randomly sampling 10% of the training set.

Table 1. Summary of benchmark datasets.

Dataset	# Train	# Test	# Feature	# Class
MNIST	$60,000$	$10,000$	784	10
Kuzushiji-MNIST	$60,000$	$10,000$	784	10
Fashion-MNIST	$60,000$	$10,000$	784	10
CIFAR-10	$50,000$	$10,000$	$3,072$	10

Table 2. Summary of real-world partial label datasets.

Dataset	# Examples	# Features	# Class	Avg # CLs	Task Domain
Lost	$1,122$	108	16	2.23	Automatic face naming
BirdSong	$4,998$	38	13	2.18	Bird song classification
MSRCv2	$1,758$	48	23	3.16	Object classification
Soccer Player	$17,472$	279	171	2.09	Automatic face naming
Yahoo! News	$22,991$	163	219	1.91	Automatic face naming

The mini-batch size is set as 256 and the number of epochs is set as 250. They all apply the cross-entropy loss function to build the partial label loss function.

B.3. Details of Architecture

In this section, we list the architecture of three models, linear, MLP, and ConvNet. The linear model is a linear-in-input model: $d - 10$. MLP refers to a 5-layer fully connected networks with ReLU as the activation function, whose architecture is $d - 300 - 300 - 300 - 300 - 10$. Batch normalization was applied before hidden layers. For both models, the softmax function was applied to the output layer, and ℓ_2-regularization was added.

The detailed architecture of ConvNet (Laine & Aila, 2016) is as follows.

0th (input) layer: (32*32*3)-

1st to 4th layers: [C(3*3, 128)]*3-Max Pooling-

5th to 8th layers: [C(3*3, 256)]*3-Max Pooling-

9th to 11th layers: C(3*3, 512)-C(3*3, 256)-C(3*3, 128)-

12th layers: Average Pooling-10

where C(3*3, 128) means 128 channels of 3*3 convolutions followed by Leaky-ReLU (LReLU) active function, $[\cdot] * 3$ means 3 such layers, etc.

B.4. Matrix Representations of Alternative Data Generations

Case 1: Each true label has a unique similar label with probability $q_1 > 0$ to enter the partial label set, while all other labels are not partial labels. When $q_1 = 0.5$, the data generation corresponds to the one proposed in (Lv et al., 2020). A matrix representation is

$$\begin{bmatrix} 1 & q_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & q_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & q_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & q_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & q_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & q_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & q_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & q_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & q_1 \\ q_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

where the element in the i-th row and the j-th column represents the conditional probability $\mathrm{P}(j \in \vec{Y} \,|\, Y = i, x)$.

Case 2: Each true label has two similar labels with probability $q_1 > 0$ to be partial labels, while all other labels are not partial labels. Here we let $q_1 = 0.3$. A matrix representation is

$$\begin{bmatrix} 1 & q_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_1 \\ q_1 & 1 & q_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & q_1 & 1 & q_1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_1 & 1 & q_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_1 & 1 & q_1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_1 & 1 & q_1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & q_1 & 1 & q_1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & q_1 & 1 & q_1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_1 & 1 & q_1 \\ q_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & q_1 & 1 \end{bmatrix}$$

Case 3: In this case, we allow more pairs of similar labels. For each true label, there exist a pair of most similar labels with probability q_1 to be partial labels, two pairs of less similar labels with probabilities q_2 and q_3 respectively. Assume that $q_1 > q_2 > q_3 > 0$. Other labels are taken as non-partial

labels. We let $q_1 = 0.5$, $q_2 = 0.3$, $q_3 = 0.1$. A matrix representation is

$$\begin{bmatrix}
1 & q_1 & q_2 & q_3 & 0 & 0 & 0 & q_3 & q_2 & q_1 \\
q_1 & 1 & q_1 & q_2 & q_3 & 0 & 0 & 0 & q_3 & q_2 \\
q_2 & q_1 & 1 & q_1 & q_2 & q_3 & 0 & 0 & 0 & q_3 \\
q_3 & q_2 & q_1 & 1 & q_1 & q_2 & q_3 & 0 & 0 & 0 \\
0 & q_3 & q_2 & q_1 & 1 & q_1 & q_2 & q_3 & 0 & 0 \\
0 & 0 & q_3 & q_2 & q_1 & 1 & q_1 & q_2 & q_3 & 0 \\
0 & 0 & 0 & q_3 & q_2 & q_1 & 1 & q_1 & q_2 & q_3 \\
q_3 & 0 & 0 & 0 & q_3 & q_2 & q_1 & 1 & q_1 & q_2 \\
q_2 & q_3 & 0 & 0 & 0 & q_3 & q_2 & q_1 & 1 & q_1 \\
q_1 & q_2 & q_3 & 0 & 0 & 0 & q_3 & q_2 & q_1 & 1
\end{bmatrix}$$

References

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.

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*, 2020.

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.

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.

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

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

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

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

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.