Leveraged Weighted Loss for Partial Label Learning

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

[1]Peking University,
Beijing, China
[2]University of Twente,
Enschede, Netherlands
[3]Samsung Research,
Beijing, China
[4]Pazhou Lab,
Guangzhou, China

June 18, 2021

Outline

Background

Background

- Labeling is labor-intensive and costly.
- True label is sometimes hard to achieve due to privacy issues.

Learning from Partial Labels

- Input variable $X \in \mathcal{X}$ is associated with a set of potential labels $\vec{Y} \in \vec{\mathcal{Y}}$.
- Find *truth label* Y for input X through observing the *partial label set* $\vec{Y}$.
- True label Y of an instance X always in the partial label set $\vec{Y}$.

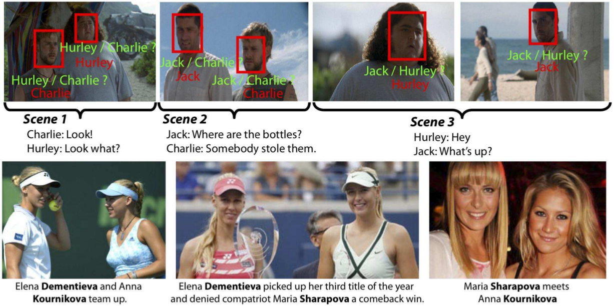


Figure: Figure 1 from *Cour et.al. 2011, Learning from Partial Labels*.

Outline

Contribution

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

Outline

Leveraged Weighted (LW) Loss Function

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

- **A binary loss function** $\psi(\cdot) : \mathcal{Y} \times \mathbb{R} \to \mathbb{R}^+$**.**
  Larger g_z for partial labels, smaller g_z for complementary labels
- **Weighting parameters** $w_z \geq 0$ **on** $\psi(g_z)$**.**
  Assign more weights to the loss of labels that are more likely to be the true.
- **Leverage parameter** $\beta \geq 0$**.**
  Larger β quickly rules out non-partial labels during training.
  It also lessens the weights assigned to partial labels.

Leveraged Weighted (LW) Loss Function

Some special cases include

1) $\beta = 0$, Jin & Ghahramani (2002)

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

2) $\beta = 0$, Lv et al. (2020)

$$\psi(\max_{y \in \boldsymbol{y}} g_y(x)) = \min_{y \in \boldsymbol{y}} \psi(g_y(x)).$$

3) $\beta = 1$, Cour et al. (2011)

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

Outline

Generalizing the Uniform Sampling Assumption

Uniform Sampling Assumption (Feng et al., 2020)

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

Intuition: if no information of $\vec{Z}$ is given, randomly guess with even probabilities whether the correct y is included in an unknown label set $\vec{Z}$ or not.

In this paper, we allow the sampling probability to be label-specific.

- Denote $q_z := \mathrm{P}(z \in \vec{Y} \mid Y = y, x) < 1$.
- $\mathrm{P}(\vec{Y} = \vec{y} \mid Y = y, x) = \prod_{s \in \vec{y}, s \neq y} q_s \cdot \prod_{t \notin \vec{y}} (1 - q_t)$.
- If rule out $\vec{Y} = [k]$,
  $\mathrm{P}(\vec{Y} = \vec{y} \mid Y = y, x) = \frac{1}{1-M} \prod_{s \in \vec{y}, s \neq y} q_s \cdot \prod_{t \notin \vec{y}} (1 - q_t)$, $M = \prod_{z \neq y} q_z$.

Generalizing the Uniform Sampling Assumption

Mule (true label) - Probability equal to 1

Donkey (false label) - High probability

Horse (false label) - High probability

Cat (false label) - Low probability

Dog (false label) - Low probability

......

- We allow the probability of each label $z \neq y$ being in the partial label set to be different.
  e.g. when the true label is *mule*, *donkey* is more likely to be picked as a partial label than *cat*.

Risk-consistent and Bayes-consistent Loss Function

Theorem

The LW partial loss function is risk-consistent with respect to the supervised loss function with the form

$$\mathcal{L}_{\psi}(y, g(x)) = w_y \psi(g_y(x)) + \sum_{z \neq y} w_z q_z \left[\psi(g_z(x)) + \beta\psi(-g_z(x))\right].$$

Theorem

Let $\mathcal{L}_{0\text{-}1}$ be the multi-class 0-1 loss. Assume that $\psi(\cdot)$ is differentiable and symmetric. For $\beta > 0$, if there exist a sequence of functions $\{\hat{g}_n\}$ such that

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

then we have

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

- $\beta > 0$, optimizing the LW loss results in the Bayes classifier under 0-1 loss.

Guidance of β

Symmetric binary loss $\psi(\cdot)$, e.g. zero-one loss, Sigmoid loss, Ramp loss, etc.

$$\mathcal{L}_\psi(y, g(x)) = w_y \psi(g_y(x)) + (\beta - 1) \sum_{z \neq y} w_z q_z \psi(-g_z(x)) + \sum_{z \neq y} w_z q_z.$$

- $\beta < 1$
  Positive weights to the untrue labels, leading to false identification
- $\beta > 1$
  Identify the true label, rule out the untrue labels;
  Corresponds to the *one-versus-all* (OVA) loss

Outline

1. Introduction
   - Background
   - Contribution
2. Methodology
   - Leveraged Weighted (LW) Loss Function
   - Theoretical Interpretations
   - Main Algorithm
3. Experiments

Main Algorithm

Input: Training data $D_n := \{(x_1, \boldsymbol{y}_1), \ldots, (x_n, \boldsymbol{y}_n)\}$;
Number of Training Epochs T;
Learning rate $\rho > 0$;
For $i = 1, \ldots, n$ initialize $w_{z,i}^{(0)} = \frac{1}{\#|\boldsymbol{y}_i|}$ for $z \in \boldsymbol{y}_i$ and $w_{z,i}^{(0)} = \frac{1}{K - \#|\boldsymbol{y}_i|}$ for $z \notin \boldsymbol{y}_i$.

for $t = 1$ **to** T **do**
  Calculate empirical risk $\bar{\mathcal{R}}_{D_n}^{(t)}(\bar{\mathcal{L}}^{(t-1)}, g(x; \theta^{(t-1)}))$;
  Update parameter $\theta^{(t)}$ for score functions and achieve $g(x; \theta^{(t)})$.
  Update weighting parameters $w_{z,i}^{(t)}$ by **respective normalization**;
end for

Output: Decision function achieved by $\hat{y} = \arg\min_{z \in [K]} g_z(x; \theta^{(T)})$.

Respective normalization: Respectively normalize the score functions $g_z(x; \theta)$ for $z \in \vec{y}$ and those for $z \notin \vec{y}$.

- Focus on the true label, rule out the most confusing non-partial label.
- Avoid partial labels out-weighting non-partial ones.

Outline

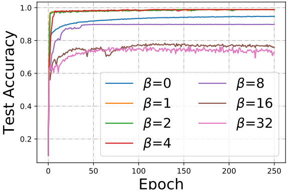
Experimental Comparisons

Table: Accuracy comparisons on benchmark datasets.

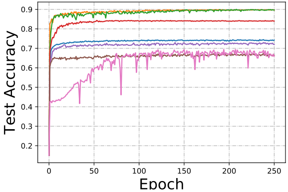
Dataset	Method	Base Model	$q = 0.1$	$q = 0.3$	$q = 0.5$
MNIST	RC	MLP	$98.44 \pm 0.11\% *$	$98.29 \pm 0.05\% *$	$98.14 \pm 0.03\% *$
	CC	MLP	$98.56 \pm 0.06\% *$	$98.32 \pm 0.06\% *$	$98.21 \pm 0.07\% *$
	PRODEN	MLP	$98.57 \pm 0.07\% *$	$98.48 \pm 0.10\% *$	$98.40 \pm 0.15\% *$
	LW-Sigmoid	MLP	$\underline{98.82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

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

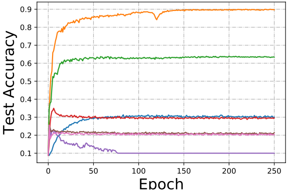
Study of β



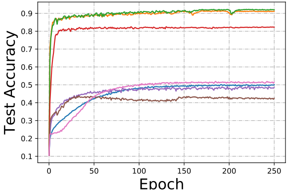
(a) MNIST, $q = 0.1$.



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



(c) CIFAR-10, $q = 0.3$.



(d) Kuzushiji-MNIST, $q = 0.5$.

Figure: Study of the leverage parameter β for LW loss.

Alternative Data Generation

Table: Accuracy comparisons with different data generation.

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

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


thank
you