ALIM: Adjusting Label Importance Mechanism for Noisy Partial Label Learning

Mingyu Xu, Zheng Lian,
Lei Feng, Bin Liu, Jianhua Tao

Introduction

--Partial Label Learning: Each example has a candidate labels set S, and the he ground-truth label *must be* in the candidate label set.

However, this assumption may not be satisfied due to the unprofessional judgment of annotators.

--Noisy Partial Label Learning[1]: Each example has a candidate labels set S, and the he ground-truth label *may not be* in the candidate label set.

Methodology

Motivation:
~ When we are unfamiliar with a test, we believe that the correct answer must be in the candidate set. Even if every option is wrong, we still choose the most likely answer.
~ As we become more familiar with the test, we learn to question the correctness of the candidate set. If we believe every option is wrong, we will consider answers outside the candidate set.
→ Adjust the importance between the candidate label set and the prediction

Definition:
S(x): Vectorized candidate label set of sample x.
P(x): Softmax probabilities of sample x.
W(x): Pseudo label of sample x.
λ: control the reliability of the candidate set

$$\tilde{S}(x) = S(x) + \lambda\left(1 - S(x)\right),$$
$$w(x) = \text{Normalize}\left(\tilde{S}(x)P(x)\right).$$

Normalize:
~Onehot: set the maximum value to 1 and others to 0.

~Scale: introduces a scaling factor K > 0 and normalizes the probabilities as follows:

$$\text{Scale}(z) = \left\{ \frac{z_i^{1/K}}{\sum_j z_j^{1/K}} \right\}_{i=1}^{c}$$

Mark:
~λ = 0 means that we fully trust the given candidate set S(x); λ = 1 means that we don't believe the candidate set but trust our own judgment P (x)
~When λ = 0 and K = 1, we can find that the classic method PRODEN[2] in the PLL is a special case of ALIM.

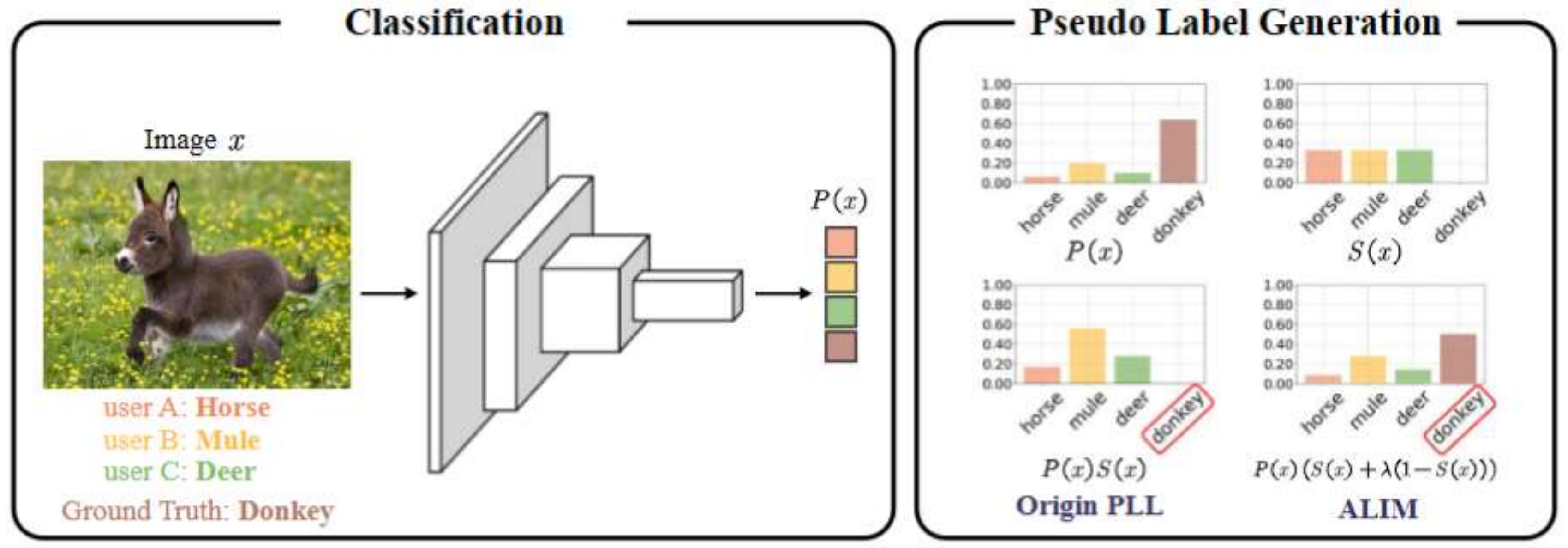


Figure 1: The core structure of ALIM. The network receives an input x and produces softmax prediction probabilities $P(x)$. Different from traditional PLL that fully trusts the candidate set, our method can deal with noisy samples through a weighting mechanism.

Analysis

~Interpretation from Objective Functions

Let P_i, S_i, and w_i be abbreviations for $P_i(x)$, $S_i(x)$, and $w_i(x)$, respectively. During training, we should optimize the following objectives:

- Minimize the classification loss on $w(x)$ and $P(x)$.
- $w(x)$ should be small at non-candidate labels.
- Entropy regularization on $w(x)$ to avoid overconfidence of pseudo labels.
- $w(x)$ should satisfy $0 \leq w_i \leq 1$ and $\sum_{i=1}^{c} w_i = 1$.

Then, the final objective function is calculated as follows:

$$\max \sum_{i=1}^{c} w_i \log P_i + M\left(\sum_{i=1}^{c} w_i S_i - 1\right) - K \sum_{i=1}^{c} w_i \log w_i$$
$$s.t. \sum_{i}^{c} w_i = 1, w_i \geq 0,$$

where M and K are penalty factors. By using Lagrange multipliers, we can observe that the penalty factor K is different for two normalization functions: $K = 0$ for Onehot$(\cdot)$ and $K > 0$ for Scale$(\cdot)$. The penalty factor M has a strong correlation with the weighting coefficient λ i.e., $\lambda = e^{-M}$. Larger M (or smaller λ) means that we have tighter constraints on $(\sum_{i=1}^{c} w_i S_i - 1)$, and therefore we should trust the given candidate set more.

~Interpretation from EM Perspective

Assumption 1 *In noisy PLL, the ground-truth label may not be in the candidate set $S(x)$. We assume that each candidate label $\{i|S_i(x) = 1\}$ has an equal probability $\alpha(x)$ of generating $S(x)$ and each non-candidate label $\{i|S_i(x) = 0\}$ has an equal probability $\beta(x)$ of generating $S(x)$.*

Besides the interpretation from objective functions, we further explain ALIM from the EM perspective We prove that the E-step aims to predict the ground-truth label for each sample and the M-step aims to minimize the classification loss. Meanwhile, ALIM is a simplified version of the results derived from EM. Specifically, EM uses an instance-dependent $\lambda(x) = \beta(x)/\alpha(x)$, while ALIM uses a global λ.

~Adaptively Adjusted Strategy (Optional)

The estimated noise level is represented as η. we prove that the value of Eq. 6 can be viewed as a metric, and the η-quantile of this value can be treated as the adaptively adjusted λ.

$$\left\{ \frac{\max_i S_i(x)P_i(x)}{\max_i (1 - S_i(x))P_i(x)} \right\}_{x \in \mathcal{D}}. \tag{6}$$

We further present results without noise rate estimation and manually adjust λ as a hyper-parameter. Through experimental analysis, this approach can also achieve competitive performance. Therefore, the adaptively adjusted strategy is optional. Its main advantage is to reduce manual efforts in hyper-parameter tuning and realize a more automatic approach for noisy PLL.

Experiments

-- ALIM achieved SOTA under noisy PLL conditions.

Table 1: Performance of different methods. ◇ denotes the models without mixup training, and ♡ denotes the models with mixup training. By default, we combine ALIM with PiCO for noisy PLL.

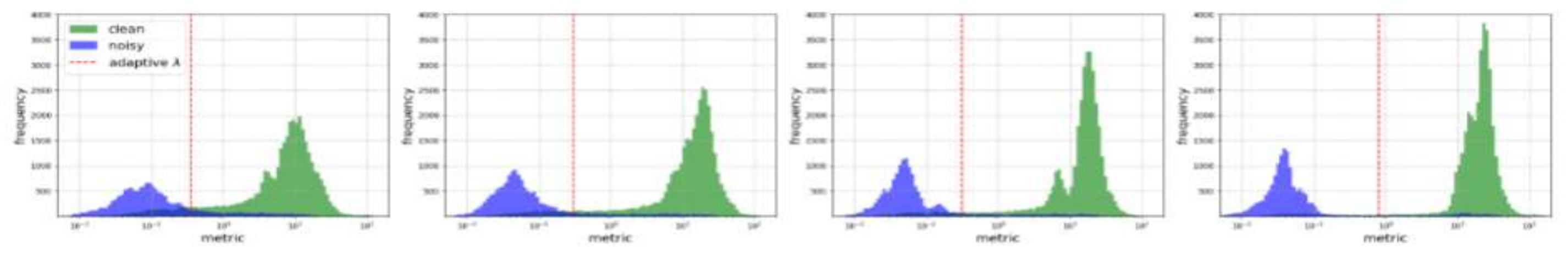
CIFAR-10	$q = 0.1$			$q = 0.3$			$q = 0.5$		
	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$
◇CC	79.81±0.22	77.06±0.18	73.87±0.31	74.09±0.60	71.43±0.56	68.08±1.12	69.87±0.94	59.35±0.22	48.93±0.52
◇RC	80.87±0.30	78.22±0.23	75.24±0.17	79.69±0.37	75.69±0.63	71.01±0.54	72.46±1.51	59.72±0.42	49.74±0.70
◇LWC	79.13±0.53	76.15±0.46	74.17±0.48	77.47±0.56	74.02±0.35	69.10±0.59	70.59±1.34	57.42±1.14	48.93±0.37
◇LWS	82.97±0.24	79.46±0.09	74.28±0.79	80.93±0.28	76.07±0.38	69.70±0.72	70.41±2.68	58.26±0.28	39.42±3.09
◇PiCO	90.78±0.24	87.27±0.11	84.96±0.12	89.71±0.18	85.78±0.23	82.25±0.32	88.11±0.29	82.41±0.30	68.75±2.62
◇CRDPLL	93.48±0.17	89.13±0.39	86.19±0.48	92.73±0.19	86.96±0.21	83.40±0.14	91.10±0.07	82.30±0.46	73.78±0.55
◇PiCO+	93.64±0.19	93.13±0.26	92.18±0.38	92.32±0.08	92.22±0.01	89.95±0.19	91.07±0.02	89.68±0.01	84.08±0.42
◇IRNet	93.44±0.21	92.57±0.25	92.38±0.21	92.81±0.19	92.18±0.18	91.35±0.08	91.51±0.05	90.76±0.10	86.19±0.41
◇ALIM-Scale	**94.15±0.14**	93.41±0.04	93.28±0.08	93.40±0.03	92.69±0.11	92.01±0.19	92.52±0.12	90.92±0.10	86.51±0.21
◇ALIM-Onehot	**94.15±0.15**	**94.04±0.16**	**93.77±0.27**	**93.44±0.16**	**93.25±0.08**	**92.42±0.17**	**92.67±0.12**	**91.83±0.08**	**89.80±0.38**
♡PiCO+	94.58±0.02	94.74±0.13	94.43±0.19	94.02±0.03	94.03±0.01	92.94±0.24	93.56±0.08	92.65±0.26	88.21±0.37
♡ALIM-Scale	95.71±0.01	95.50±0.08	95.35±0.13	95.31±0.16	94.77±0.07	94.36±0.03	94.71±0.04	93.82±0.13	90.63±0.10
♡ALIM-Onehot	**95.83±0.13**	**95.86±0.15**	**95.75±0.19**	**95.52±0.15**	**95.41±0.13**	**94.67±0.21**	**95.19±0.24**	**93.89±0.21**	**92.26±0.29**

--ALIM always brings performance improvement under noisy conditions.

Table 2: Compatibility of ALIM on different PLL methods.

PLL	ALIM	CIFAR-10								
		$q = 0.1$			$q = 0.3$			$q = 0.5$		
		$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$
◇RC	×	80.87±0.30	78.22±0.23	75.24±0.17	79.69±0.37	75.69±0.63	71.01±0.54	72.46±1.51	59.72±0.42	49.74±0.70
◇RC	√	88.81±0.17	87.16±0.20	85.53±0.05	86.21±0.17	83.64±0.07	79.83±0.43	77.40±0.31	69.13±0.71	56.75±1.59
◇PiCO	×	90.78±0.24	87.27±0.11	84.96±0.12	89.71±0.18	85.78±0.23	82.25±0.32	88.11±0.29	82.41±0.30	68.75±2.62
◇PiCO	√	94.15±0.15	94.04±0.16	93.77±0.27	93.44±0.16	93.25±0.08	92.42±0.17	92.67±0.12	91.83±0.08	89.80±0.38
◇CRDPLL	×	93.48±0.17	89.13±0.39	86.19±0.48	92.73±0.19	86.96±0.21	83.40±0.14	91.10±0.07	82.30±0.46	73.78±0.55
◇CRDPLL	√	96.03±0.23	95.01±0.32	93.36±0.10	95.32±0.13	93.27±0.29	91.20±0.06	93.82±0.05	90.20±0.04	84.24±0.28
PLL	ALIM	CIFAR-100								
		$q = 0.01$			$q = 0.03$			$q = 0.05$		
		$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$	$\eta = 0.1$	$\eta = 0.2$	$\eta = 0.3$
◇RC	×	52.73±1.05	48.59±1.04	45.77±0.31	52.15±0.19	48.25±0.38	43.92±0.37	46.62±0.34	45.46±0.21	40.31±0.55
◇RC	√	61.46±0.26	60.10±0.23	55.67±0.28	57.43±0.20	52.98±0.27	48.74±0.37	56.40±0.60	51.91±0.12	46.87±0.74
◇PiCO	×	68.27±0.08	62.24±0.31	58.97±0.09	67.38±0.09	62.01±0.33	58.64±0.28	67.52±0.43	61.52±0.28	58.18±0.65
◇PiCO	√	72.26±0.23	71.98±0.29	71.04±0.31	71.43±0.21	70.79±0.43	70.14±0.25	72.28±0.28	70.60±0.44	70.05±0.43
◇CRDPLL	×	68.12±0.13	65.32±0.34	62.94±0.28	67.53±0.07	64.29±0.27	61.79±0.11	67.17±0.04	64.11±0.42	61.03±0.43
◇CRDPLL	√	69.98±0.30	68.58±0.16	66.90±0.16	69.60±0.20	67.67±0.22	66.15±0.12	68.75±0.06	67.07±0.29	64.69±0.23

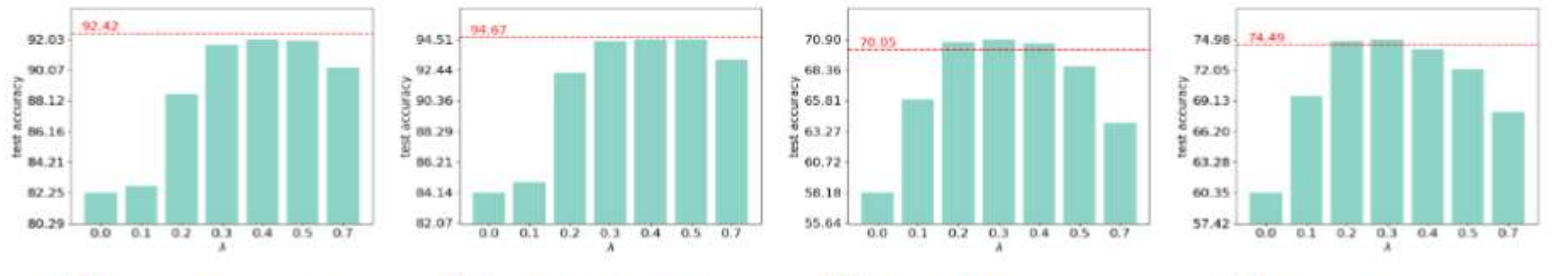
--ALIM's adaptively adjusted λ serves as a suitable boundary for clean and noisy subsets



(a) 100 epoch (b) 200 epoch (c) 400 epoch (d) 800 epoch

Figure 2: Distribution of the value in Eq. 6 for clean and noise subsets with increasing training iterations. We conduct experiments on CIFAR-10 ($q = 0.3, \eta = 0.3$) with $e_0 = 80$.

--ALIM's adaptively adjusted λ reduce manual efforts in hyper-parameter tunining



(a) CIFAR-10 (w/o mixup) (b) CIFAR-10 (w/ mixup) (c) CIFAR-100 (w/o mixup) (d) CIFAR-100 (w/ mixup)

Figure 3: Classification performance of manually adjusted λ and adaptively adjusted λ. We conduct experiments on CIFAR-10 ($q = 0.3, \eta = 0.3$) and CIFAR-100 ($q = 0.05, \eta = 0.3$). We mark the results of the adaptively adjusted strategy with red lines.

References

[1] Lv, Jiaqi, Biao Liu, Lei Feng, Ning Xu, Miao Xu, Bo An, Gang Niu, Xin Geng, and Masashi Sugiyama. "On the robustness of average losses for partial-label learning." IEEE Transactions on Pattern Analysis and Machine Intelligence (2023)

[2] Lei Feng, Jiaqi Lv, Bo Han, Miao Xu, Gang Niu, Xin Geng, Bo An, and Masashi Sugiyama. Provably consistent partial-label learning. In Proceedings of the Advances in Neural Information Processing Systems, pages 10948–10960, 2020.

Code: https://github.com/zeroQiaoba/ALIM