Progressive Identification of True Labels for Partial-Label Learning

Jiaqi Lv[1] Miao Xu[2,3] Lei Feng[4] Gang Niu[2] Xin Geng[1] Masashi Sugiyama[2,5]

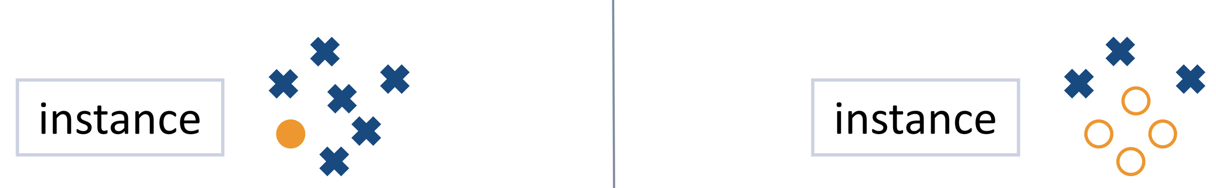
In: Proceedings of the 37th International Conference on Machine Learning (ICML 2020)

1 School of Computer Science and Engineering, Southeast University, Nanjing, China
2 RIKEN Center for Advanced Intelligence Project, Tokyo, Japan
3 The University of Queensland, Australia
4 School of Computer Science and Engineering, Nanyang Technological University, Singapore
5 University of Tokyo, Tokyo, Japan.

Introduction

Partial-label learning (PLL) is a typical weakly supervised learning problems, and arises in many real-world tasks

Ordinary multi-class classification (i.e., supervised learning) | Partial-Label Learning



P (true) & N labels are available for training

a set of possible P labels (candidate labels) are available for training

● positive (true) label ○ candidate label ✖ negative label

Most existing PLL methods must be solved in specific manners, making their computational complexity **a bottleneck for scaling up to big data**

Let PLL enjoy the leading-edge models and optimizers from **deep learning** communities

Let the PLL method **not benefit purely from** the network architecture, but also our careful algorithm design

Classifier-Consistent Risk Estimator

Notation

	Ordinary multi-class classification	Partial-Label Learning
Space	$\mathcal{X}\subseteq\mathbb{R}^d$ $\quad\mathcal{Y}=[c]:=\{1,2,\ldots,c\}$	$\mathcal{S}=\{\mathcal{P}(\mathcal{Y})/\emptyset/\mathcal{Y}\}$
Random variables	$(X,Y)\in\mathcal{X}\times\mathcal{Y}$	$(X,S)\in\mathcal{X}\times\mathcal{S}$
Density	$p(x,y)$	$p(x,s)$
Expectation	$\mathcal{R}(\boldsymbol{g})=\mathbb{E}_{(X,Y)\sim p(x,y)}[\ell(\boldsymbol{g}(X),\boldsymbol{e}^Y)]$	$\mathcal{R}_{\mathrm{PLL}}(\boldsymbol{g})=\mathbb{E}_{(X,S)\sim p(x,s)}[\ell_{\mathrm{PLL}}(\boldsymbol{g}(X),S)]$
Optimal classifier	$\boldsymbol{g}^*=\arg\min_{\boldsymbol{g}\in\mathcal{G}}\mathcal{R}(\boldsymbol{g})$	$\boldsymbol{g}^*_{\mathrm{PLL}}=\arg\min_{\boldsymbol{g}\in\mathcal{G}}\mathcal{R}_{\mathrm{PLL}}(\boldsymbol{g})$

$\boldsymbol{g}:\mathcal{X}\to\mathbb{R}^c \qquad g_k(X)=p(Y=k|X) \qquad \hat{Y}=\arg\max_{i\in\mathcal{Y}} g_i(X)$

Lemma 1 (Liu & Dietterich, 2014) The ambiguity degree is defined as

$$\gamma=sup_{(X,Y)\sim p(x,y),\bar{Y}\in\mathcal{Y},S\sim p(s|x,y),\bar{Y}\neq Y}\Pr(\bar{Y}\in S)$$

If $\gamma<1$, i.e. the under the **small ambiguity degree condition**, the PLL problem is **ERM learnability**.

a negative label **is not always co-occurred** with the true label

a classification error made on any instance will be detected with probability at least 1 - γ (Liu & Dietterich, 2014)

Lemma 2. If a certain loss function is used (e.g. the cross-entropy loss or mean squared error loss), the optimal classifier satisfies

$g_i^*(X)=p(Y=i|X)$

the optimal classifier $\boldsymbol{g}^*=\arg\min_{\boldsymbol{g}\in\mathcal{G}}\mathcal{R}(\boldsymbol{g})$ can **recover the class-posterior probability**

Classifier-Consistent Risk Estimator

Partial label risk estimator

only one label contributes to retrieve the classifier!

$$\ell_{\mathrm{PLL}}(\boldsymbol{g}(X),S)=\min_{i\in S}\ell(\boldsymbol{g}(X),\boldsymbol{e}^i)$$
$$\mathcal{R}_{\mathrm{PLL}}(\boldsymbol{g})=\mathbb{E}_{(X,S)\sim p(x,s)}\min_{i\in S}\ell(\boldsymbol{g}(X),\boldsymbol{e}^i)$$

Classifier-consistency

Suppose that the learning is conducted under the deterministic scenario, and Lemma 1 and Lemma 2 are satisfied. Then the optimal PLL minimizer **is equivalent to** the ordinary optimal minimizer

$$\boldsymbol{g}^*_{\mathrm{PLL}}=\boldsymbol{g}^*$$

Estimation error bound

For any δ > 0, we have with probability at least 1 – δ

$$\mathcal{R}_{\mathrm{PLL}}(\widehat{\boldsymbol{g}}_{\mathrm{PLL}})-\mathcal{R}_{\mathrm{PLL}}(\boldsymbol{g}^*_{\mathrm{PLL}})\leq 4cL_\ell\mathfrak{R}_n(\mathcal{G})+2M\sqrt{\frac{\log(2/\delta)}{2n}}$$

This means the risk of the empirical classifier learned by ERM can be bounded by the risk of the optimal PLL classifier

Benchmark Solution

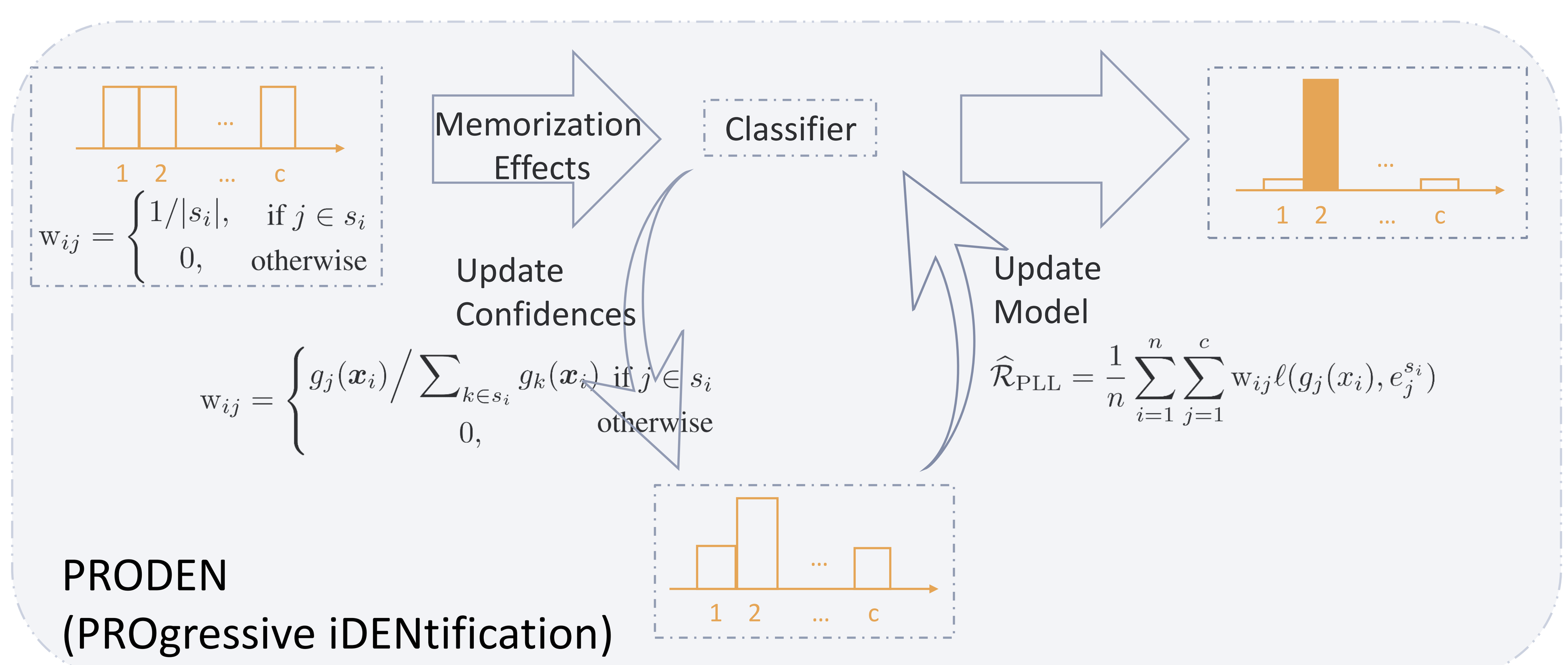
- **Difficulty:** the min operator is non-differentiable
- **Ideally:** only one (true) label should be taken into account
- **Our solution:** relax the minimal loss by the shifting confidences
- **Advantage:** this method can be easily implemented over flexible learning models and powerful stochastic optimization

Requirement on the loss function: can be decomposed onto each label:

$$\ell(\boldsymbol{g}(X),\boldsymbol{e}^Y)=\sum_{i=1}^{c}\ell(g_i(X),e_i^Y)$$

Thus **with appropriate confidences w_i**, the risk can be expressed as

$$\widehat{\mathcal{R}}_{\mathrm{PLL}}=\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{c}\mathrm{w}_{ij}\ell(g_j(x_i),e_j^{s_i}) \qquad \boldsymbol{e}^{s_i}=\sum_{k\in s_i}\boldsymbol{e}^k \qquad \mathbf{w}_i\in\Delta^{c-1}$$



Remarks

- PRODEN gets rid of the overfitting issue of EM methods
- PRODEN has great flexibility for models and loss functions

Experiments

Datasets

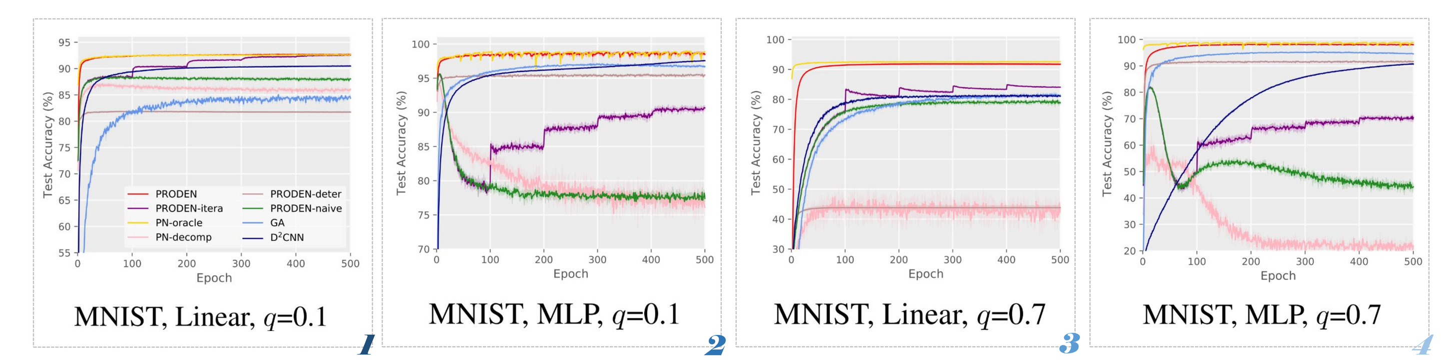
Benchmark datasets: MNIST, Fashion-MNIST, Kuzushiji-MNIST, CIFAR-10

➢ Generate partially labeled versions by **a binomial/pair flip strategy** with q: $q=\Pr(\tilde{y}=1|y=0)$

UCI datasets: Yeast, Texture, Dermatology, Synthetic Control, 20Newsgroups

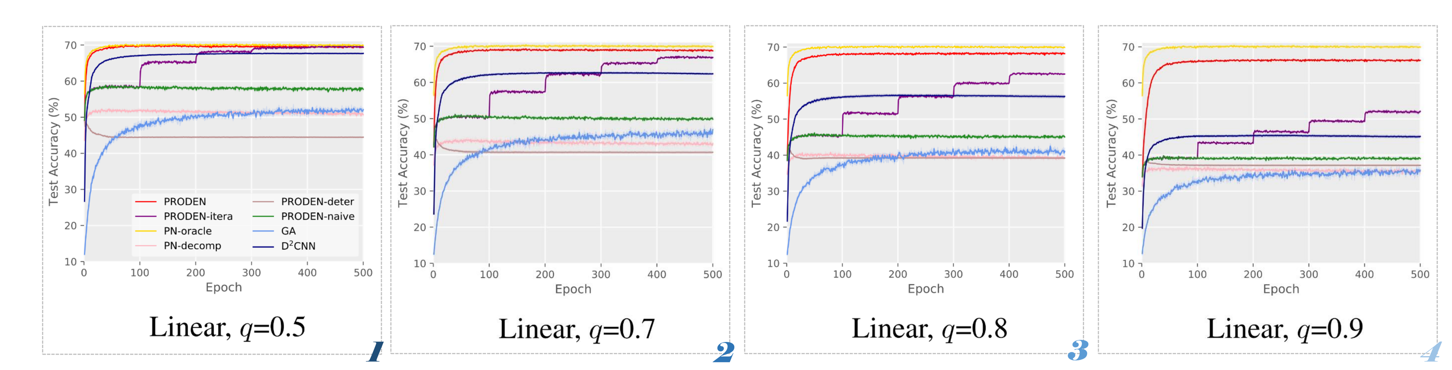
Real-world partial-label datasets: Lost, Birdsong, MSRCv2, Soccer Player, Yahoo! News

Results on MNIST in the binomial case



- **PRODEN** is **always the best** method and **comparable to** PN-oracle with all the models
- The performance of the baselines is greatly reduced with a large flipping probability
- The superiority always stands out for PRODEN compared with two deep methods GA and D²CNN

Analysis on the ambiguity degree on Kuzushiji-MNIST



- **A pair flip strategy** to simulate ambiguity degree: as $n\to\infty,\gamma\to q$
- **PRODEN** tends to be less affected with increased ambiguity

Conclusion

- We proposed a risk estimator for PLL, theoretically analyzed the classifier-consistency, and established an estimation error bound
- We proposed a method for PLL which is compatible with any learning model including DNNs or stochastic optimizer
- Experiments demonstrated our proposal is compared favorably with state-of-the-art PLL methods

More information

http://palm.seu.edu.cn
https://arxiv.org/abs/2002.08053
https://github.com/Lvcrezia77/PRODEN