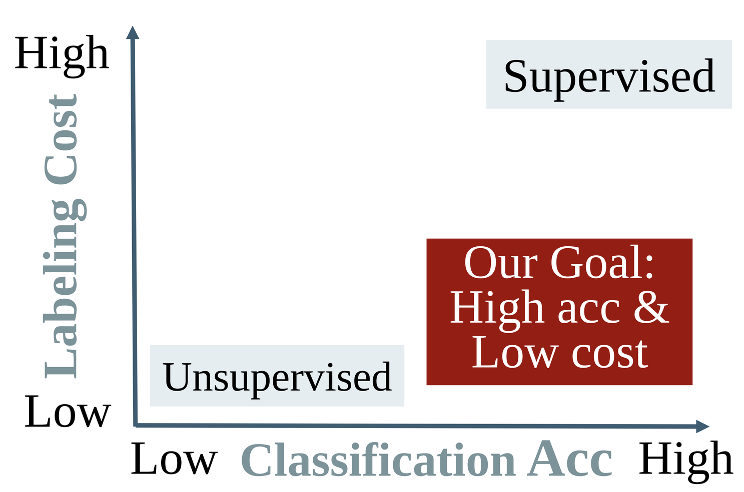
What Makes Partial-Label Learning Algorithms Effective?

東南大學 SOUTHEAST UNIVERSITY
THE UNIVERSITY OF QUEENSLAND AUSTRALIA
RIKEN AIP
東京大学
NEURAL INFORMATION PROCESSING SYSTEMS

Jiaqi Lv, Yangfan Liu, Shiyu Xia, Ning Xu, Miao Xu, Gang Niu ,
Min-Ling Zhang, Masashi Sugiyama, Xin Geng

Background



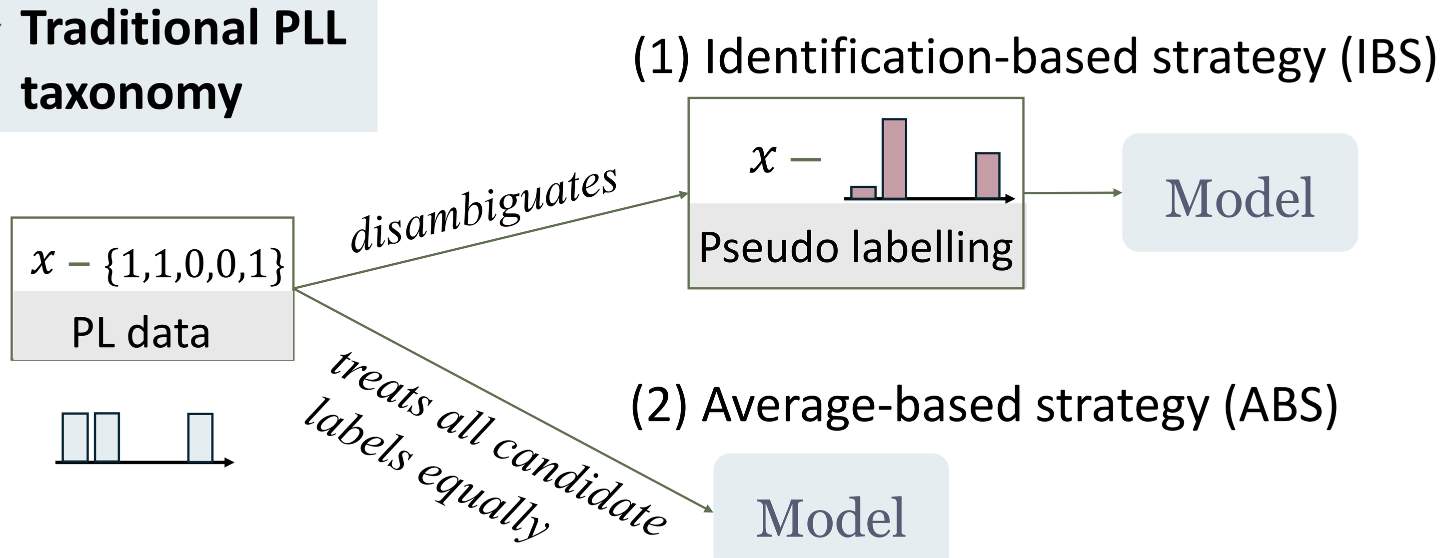
- Reducing labeling costs is a significant challenge
- Weakly supervised learning emerges as a crucial branch

Partial-Label Data:

Partial label:
Husky, Alaskan, Eurasian gray wolf

- Partial Label Learning (PLL) is a typical weakly supervised classification problem
- Each instance is labeled with a partial label, where a fixed but unknown label is true

◆ Traditional PLL taxonomy



◆ SOTA PLL methods with quite different technical routes

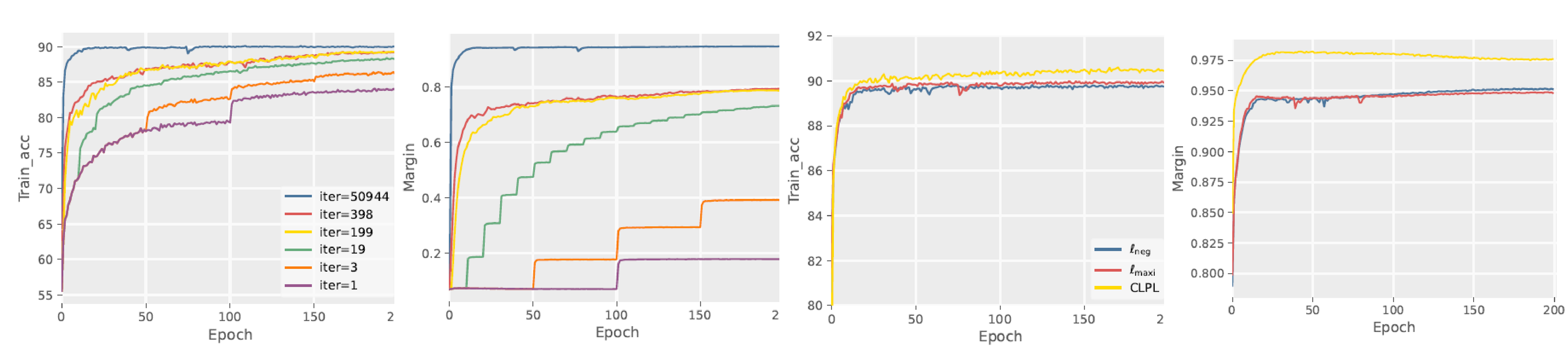
Table 1: Comparison of techniques used in eight prominent PLL methods. ✓/× indicates whether a technique is used, and an underline denotes the key components of the respective methods.

PLL methods	Mini-b. purif.	Mixup	Data augment.	Exp.Mov Average	Match DA	Match DM	Main assumption
PRODEN [25]	$\underline{\checkmark}$	×	×	×	×	$\underline{\checkmark}$	DNNs learn pattern first
CC [12]	✓	×	×	×	×	×	$\underline{\text{PLs are generated uniformly}}$
LWC [37]	✓	×	×	×	×	✓	$\underline{\text{PLs are class-dependent}}$
PiCO [34]	✓	×	✓	✓	$\underline{\checkmark}$	$\underline{\checkmark}$	Same class representations cluster
DPLL [38]	✓	×	✓	×	$\underline{\checkmark}$	×	Input is invarient to the translations
SoLar [33]	✓	✓	✓	✓	✓	×	$\underline{\text{High-confid. sample is likely correct}}$
PaPi [40]	✓	✓	✓	✓	✓	$\underline{\checkmark}$	Same class representations cluster
CroSel [30]	✓	✓	✓	✓	✓	$\underline{\checkmark}$	$\underline{\text{Stable high-confid. sample is correct}}$

Main Question: What makes PLL algorithms effective?

- Past intuition: (1) IBS is more effective; (2) Stronger tricks lead to better performance
- This understanding is vital for choosing the best direction for future algorithm design
- It also can act as a conclusive work to prevent redundant efforts in future research

Observation and Answer 1

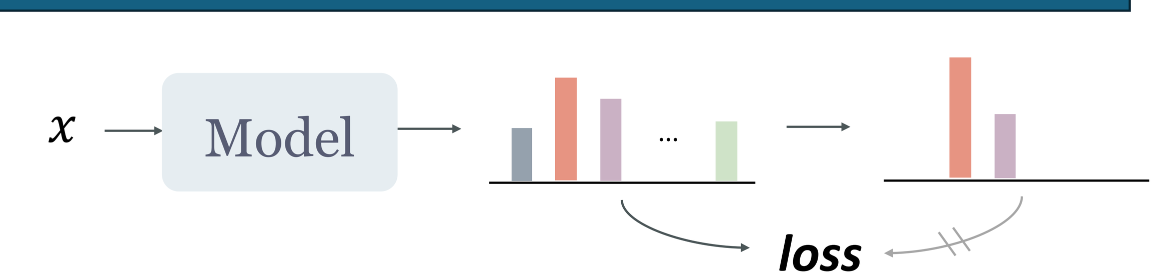


(a) Acc of IBS (b) Margin of IBS (c) Acc of ABS (d) Margin of ABS

- Observation 1: IBS and ABS often overlap in the dynamic characteristics of their implementations
- Our Answer 1: There is no need to predefine a method's category based on traditional taxonomy and then judge its utility by that classification

Observation and Answer 2

- An IBS method:



- An ABS method: optimizes $\ell_{\text{maxi}}(\boldsymbol{x}, f) = -\log \sum_{i \in S} f_i(\boldsymbol{x})$

- Observation 2: An ABS method has exactly the same optimization goals as a representative IBS algorithm
- Our Answer 2: PLL must use supervision, and supervision can be used in two ways: directly within the loss function or to manipulate pseudo-labels on-the-fly

Observation and Answer 3

Table 1: Conceptual and empirical comparisons (%) of various simplifications of DPLL. ✓/× indicates whether a technique is used.

Methods	Mini-b. purif.	ABS loss	Data augment.	DA	FMNIST 0.3	FMNIST 0.7	CIFAR-100 0.05	CIFAR-100 0.1	mini-I.Net ins.-dep.
Eq. 4+6 DPLL	✓	✓	✓	✓	93.82	**92.68**	76.81	75.93	52.22
Eq. 4+7	✓	✓	✓	✓	93.80	92.44	79.35	78.85	53.40
Eq. 8+6	✓	✓	✓	✓	93.49	92.19	**79.75**	78.87	53.78
Eq. 8+7	✓	✓	✓	✓	93.57	92.20	79.53	78.85	54.09
Eq. 6	×	✓	✓	×	93.58	92.12	76.96	75.94	44.69
Eq. 8 DASM	✓	×	✓	✓	93.89	92.85	79.70	**79.62**	54.71
DASM-H	×	×	✓	✓	93.86	92.37	78.25	33.22	34.59
DASM-S	×	×	✓	✓	93.28	90.75	78.65	76.22	36.71
DASM-E	×	×	✓	×	93.80	92.35	79.30	79.11	53.44
SASM	✓	×	✓	×	**93.83**	92.18	79.38	78.19	**55.45**
DASM w/o.aug	✓	×	×	✓	90.86	89.60	60.18	56.39	30.85

- Observation 3: By dissecting SOTA PLL algorithms with quite different technical routes, we find the common elements contributing to success
- Mini-batch PL purification is a minimal design principles in PLL
- Mini-batch PL purification is a process where for each mini-batch B selected at iteration t, the weights are updated such that the distinction among candidate labels' contributions increases over iterations:

$$w_{t+1}(\boldsymbol{x}; f, S) = g(\text{model's confidence for } \boldsymbol{x} \text{ based on current and previous iterations})$$

with g being a strictly increasing function that increases the weight for more likely candidate labels according to the model's confidence

A Step Forward: StreamPurify