模式分析与机器智能
工业和信息化部重点实验室
MIIT Key Laboratory of
Pattern Analysis & Machine Intelligence

ParNeC 模式识别与神经计算研究组
PAttern Recognition and NEural Computing

Partial Label Learning with a Partner

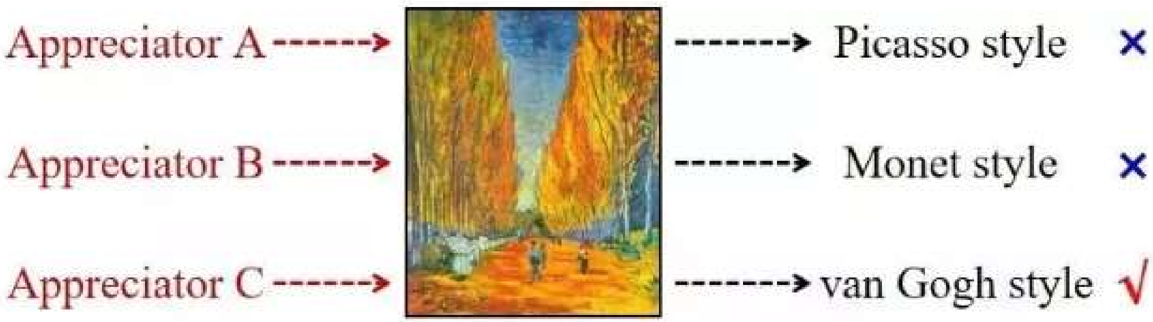
Chongjie Si[1], Zekun Jiang[1], Xuehui Wang[1], Yan Wang[2], Xiaokang Yang[1], Wei Shen[1]*

[1]MoE Key Lab of Artificial Intelligence, AI Institute, Shanghai Jiao Tong University
[2]Shanghai Key Lab of Multidimensional Information Processing, East China Normal University
{chongjiesi, zkjiangzekun.cmu, wangxuehui, xkyang, wei.shen}@sjtu.edu.cn; ywang@cee.ecnu.edu.cn

AAAI 2024

Background

Partial Label Learning (PLL)



Partial label learning (PLL), also known as superset-label learning or ambiguous label learning, is a representative weakly supervised learning framework which learns from inaccurate supervision information.

In partial label learning, each instance is associated with a set of candidate labels with only one being ground-truth and others being false positive.

As the ground-truth label of a sample conceals in the corresponding candidate label set, which can not be directly acquired during the training process, partial label learning task is a quite challenging problem.

Related work

To tackle the mentioned challenge, existing works mainly focus on disambiguation

Averaging-based approaches

Such as PL-KNN(2005) averages the candidate labels of neighboring samples to make the prediction.

Identification-based approaches

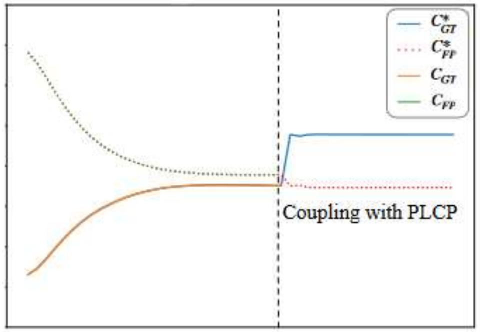
The ground-truth label is treated as a latent variable and can be identified through an iterative optimization procedure such as EM. Labeling confidence based strategy is proposed in many state-of-the-art identification based approaches for better disambiguation.

Deep-learning based models

PICO(2022) is a contrastive learning-based approach devised to tackle label ambiguity in partial label learning.
PRODEN(2020) is a model where the simultaneous updating of the model and identification of true labels are seamlessly integrated.

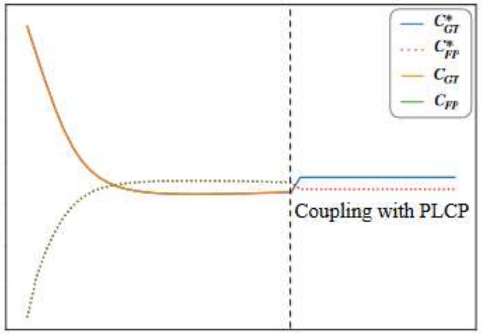
Motivation

However, a significant yet rarely studied question arises in the context of such algorithms: can a classifier correct a false positive candidate label (i.e., invalid candidate label) with a large or upward-trending labeling confidence at a later stage?



(a)

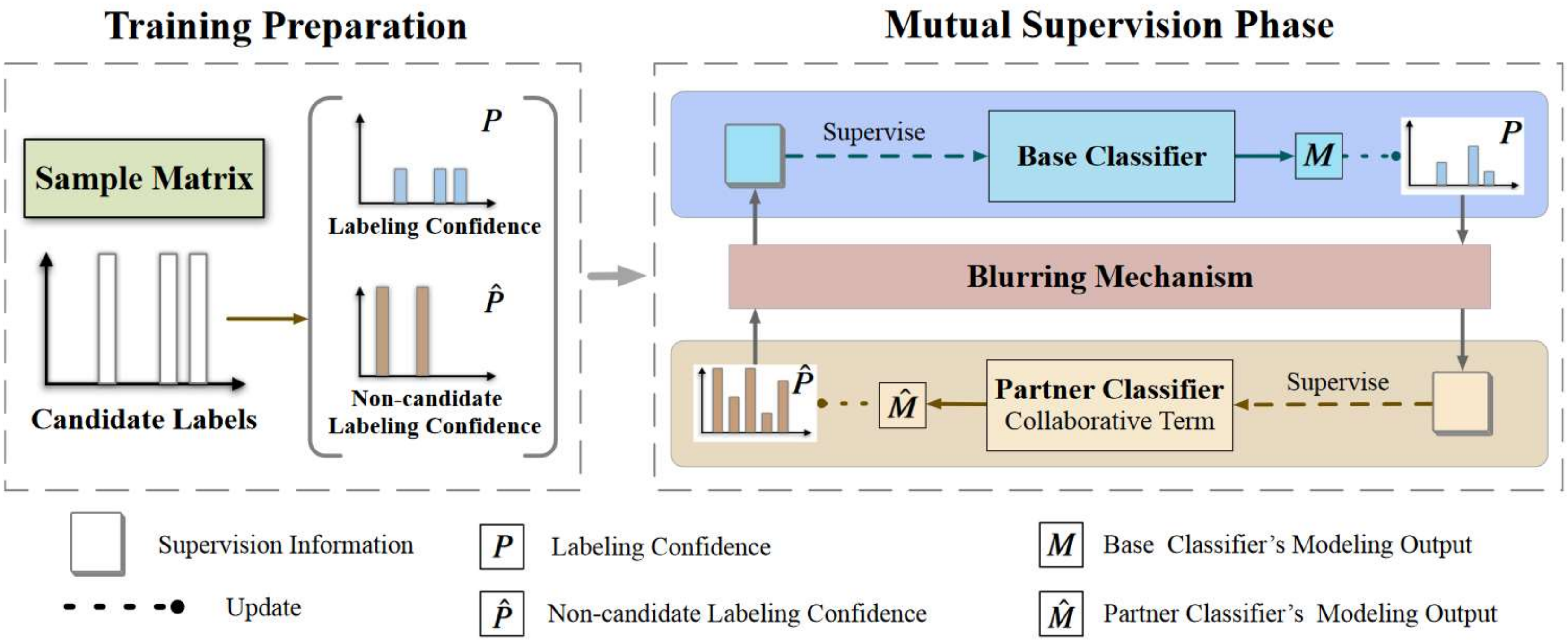
(a). For a false positive candidate label with a large labeling confidence, although its confidence may decrease properly, it could still be larger than the ground-truth one's.



(b)

(b). The labeling confidence of a false positive candidate label keeps increasing and becomes the largest, which misleads the final prediction.

Method



Denote $X = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^{n \times q}$ the sample matrix with n instances, and $Y = [y_1, y_2, \dots, y_n]^T \in \{0,1\}^{n \times l}$ the partial label matrix with l labels, where $y_{ij} = 1$ (resp. $y_{ij} = 0$) if the j-th label of x_i resides in (resp. does not reside in) its candidate label set. Given the partial label data set $\mathcal{D} = \{x_i, S_i | 1 \le i \le n\}$, the task of PLL is to learn a multi-class classifier f: $X \to Y$ based on $\mathcal{D}$.

Method

Base Classifier

Suppose $P = [p_1, p_2, \dots, p_n]^T \in \mathbb{R}^{n \times l}$ is the labeling confidence matrix.

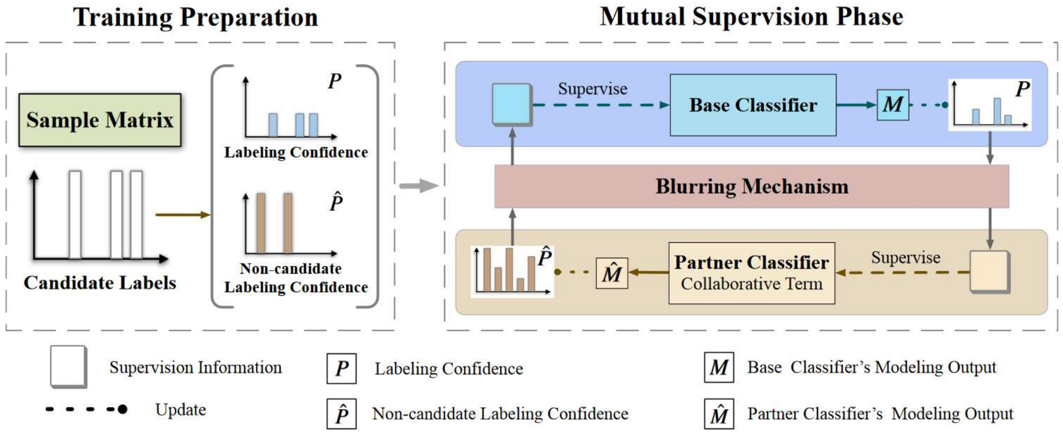
P is initialized according to the base classifier. Otherwise, it is initialized as follows:

$$p_{ij} = \begin{cases} \frac{1}{\sum_j y_{ij}} & \text{if} \quad y_{ij} = 1 \\ 0 & \text{otherwise .} \end{cases}$$

(1) $\sum_j p_{ij} = 1$,
(2) $0 \le p_{ij} \le y_{ij}$.

Denote the modeling output matrix $M = [m_1, m_2, \dots, m_n]^T \in \mathbb{R}^{n \times l}$.

P is updated to P_1 through the following equation: $\mathrm{P}_1 = \mathcal{T}_0\big(\mathcal{T}_Y(\alpha P + (1-\alpha)M)\big)$, where $\mathcal{T}_0(a) = \max\{0, a\}$, $\mathcal{T}_Y(a) = \min\{y_{ij}, a\}$.



Blurring Mechanism

$$Q_1 = \phi(e^k P_1) \odot Y$$

where $\phi(A) = \left[\exp(a_{ij})\right]_{n \times l}$, k is a temperature parameter, $\odot$ represents the Hadamard product.

Then normalize each row of Q_1 to satisfy the two constraints of labeling confidence, and output the result $O_1 \in \mathbb{R}^{n \times l}$.

Method

Partner Classifier

Denote the non-candidate label matrix $\hat{Y} = [\hat{y}_{ij}]_{n\times l}$ where $\hat{y}_{ij} = 0$ (resp. $\hat{y}_{ij} = 1$) if the j-th label is (resp. is not) in the candidate label set of x_i.

$\hat{P} = [\hat{p}_1, \hat{p}_2, \dots, \hat{p}_n]^T \in \mathbb{R}^{n\times l}$ the non-candidate labeling confidence matrix.

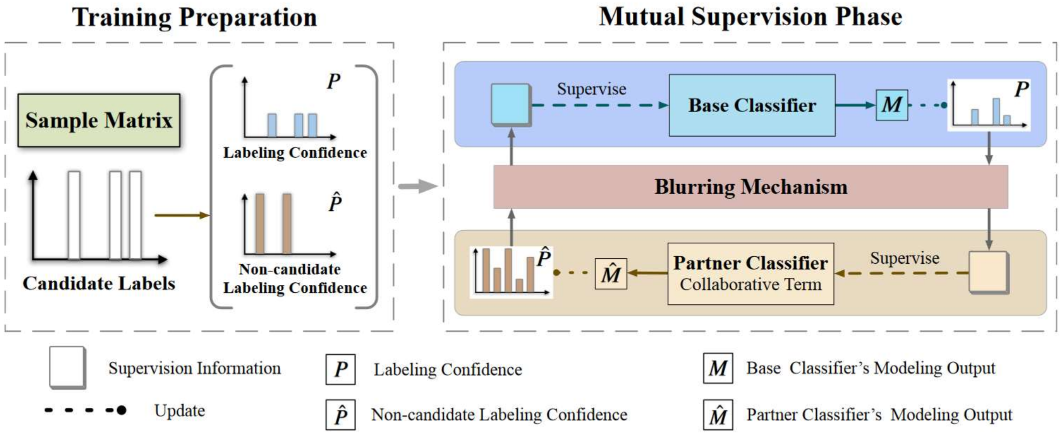
(1) $\sum_j \hat{p}_{ij} = l - 1$,

(2) $\hat{y}_{ij} \leq \hat{p}_{ij} \leq 1$.

Suppose $\widehat{W} = [\widehat{w}_1, \widehat{w}_2, \dots, \widehat{w}_q]^T \in \mathbb{R}^{q\times l}$ is the weight matrix, the partner classifier is formulated as follows:

$$\min_{\hat{\mathbf{W}},\hat{\mathbf{b}},\mathbf{C}} \quad \left\|\mathbf{X}\hat{\mathbf{W}} + \mathbf{1}_n\hat{\mathbf{b}}^{\mathsf{T}} - \mathbf{C}\right\|_F^2 + \lambda\left\|\hat{\mathbf{W}}\right\|_F^2$$

$$\text{s.t.} \quad \hat{\mathbf{Y}} \leq \mathbf{C} \leq \mathbf{1}_{n\times l}, \mathbf{C}\mathbf{1}_l = (l-1)\mathbf{1}_n,$$



where $\hat{b} = [\hat{b}_1, \hat{b}_2, \dots, \hat{b}_l]^T \in \mathbb{R}^l$ is the bias term,

$1_n \in \mathbb{R}^n$ is an all one vectors,

$1_{n\times l}$ is an all one matrix with size $n \times l$,

λ is a hyper-parameter trading off these terms,

$\|\widehat{W}\|_F$ is the Frobenius norm of the weight matrix.

$C \in \mathbb{R}^{n\times l}$ represents non-candidate labeling confidence, which is a temporary variable only used for optimization in the partner classifier.

Method

The Collaborative Term

The ideal state of p_i is one-hot.
The ideal state of $\hat{p}_i$ is zero-hot.
So, the smallest value of $p_i^T \hat{p}_i$ is obtained when $\hat{p}_i$ is zero-hot (p_i is one-hot).

$$\min_{\hat{\mathbf{W}},\hat{\mathbf{b}},\mathbf{C}} \left\|\mathbf{X}\hat{\mathbf{W}} + \mathbf{1}_n \hat{\mathbf{b}}^\mathrm{T} - \mathbf{C}\right\|_F^2 + \boxed{\gamma \mathrm{tr}\left(\mathbf{O}_1 \mathbf{C}^\mathrm{T}\right)} + \lambda \left\|\hat{\mathbf{W}}\right\|_F^2$$

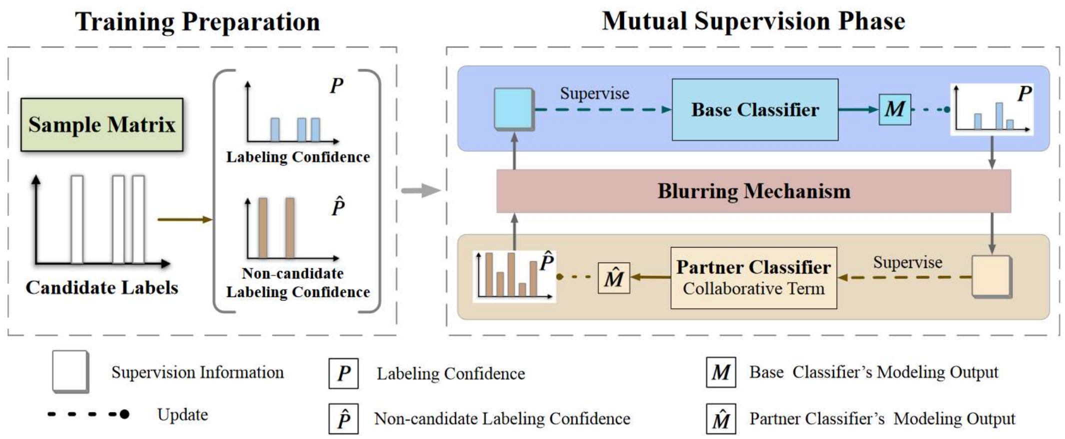
$$\text{s.t.} \quad \hat{\mathbf{Y}} \le \mathbf{C} \le \mathbf{1}_{n\times l}, \mathbf{C}\mathbf{1}_l = (l-1)\mathbf{1}_n,$$

where γ is a hyper-parameter.

The problem can be solved via an alternative and iterative manner.

The modeling output $\widehat{M}$ for the training data is

$$\widehat{M} = \mathrm{X}\widehat{\mathrm{W}} + 1_n \hat{b}^T.$$



The non-candidate labeling confidence $\hat{P}$ is updated to $\hat{P}_1$ following

$$\hat{P}_1 = \mathcal{T}_1\left(\mathcal{T}_{\hat{Y}}\left(\alpha\hat{P} + (1-\alpha)\widehat{M}\right)\right)$$

where $\mathcal{T}_1(m) = \min\{1, m\}$, $\mathcal{T}_{\hat{Y}}(m) = \max\{\hat{y}_{ij}, m\}$.

Then get $\hat{Q}_1 = \phi\left(e^k\left(1 - \hat{P}_1\right)\right) \odot Y$ and normalize to $\hat{O}_1$.

Method

Update $\widehat{W}$ and $\hat{b}$

With C fixed, the problem w.r.t. $\widehat{W}$ and $\hat{b}$ can be written as

$$\min_{\widehat{W},\hat{b}} \left\| X\widehat{W} + 1_n\hat{b}^T - C \right\|_F^2 + \lambda \left\| \widehat{W} \right\|_F^2$$

which is a least square problem with the closed-form solution as

$$\widehat{W} = (X^T X + \lambda I_{n\times n}) X^T C$$

$$\hat{b} = \frac{1}{n}\left(C^T 1_n - \widehat{W}^T X^T 1_n\right)$$

where $I_{n\times n}$ is the identity matrix with the size $n \times n$.

Update C

With Ŵ and $\hat{b}$ fixed, the C-subproblem can be formulated as

$$\min_{C} \left\| X\widehat{W} + 1_n\hat{b}^T - C \right\|_F^2 + \gamma tr(O_1 C^T)$$

$$s.t. \hat{Y} \le C \le 1_{n\times l}, C1_l = (l-1)1_n$$

For simplicity, O_1 is written as O and $J = X\widehat{W} + 1_n\hat{b}^T$.

Notice that each row of C is independent to other rows, therefore the problem can be solved row by row:

$$\min_{C_i} C_i^T C_i + (\gamma O_i - 2J_i)^T C_i$$

$$s.t. \hat{Y}_i \le C_i \le 1_n, C_i 1_l = l - 1$$

The problem is a standard Quadratic Programming (QP) problem, which can be solved by off-the-shelf QP tools.

Method

Kernel Extension

Denote $\phi(\cdot): \mathbb{R}^q \rightarrow \mathbb{R}^h$ the feature mapping that maps the feature space to some higher dimension space with h dimensions.

Then we can rewrite as

$$\min_{\widehat{W},\hat{b}} \|Z\|_F^2 + \lambda \left\|\widehat{W}\right\|_F^2$$

$$s.t. Z = \Phi\widehat{W} + 1_n \hat{b}^T - C$$

where $\Phi = [\phi(x_1), \phi(x_2), \dots, \phi(x_n)]$.

Deep-learning Extension

Denote a model in $\mathcal{B}$ with such architecture $g(\cdot)$, specifically, an additional model $\hat{g}(\cdot)$ with the same architecture as $g(\cdot)$ is introduced as the partner classifier, which predicts the non-candidate labeling confidence of each sample.

$$\mathcal{L}_{com} = -\sum_{i=1}^{l} (1 - y_i) \log(\hat{g}_i(x))$$

$$\mathcal{L}_{col} = -\sum_{i=1}^{l} p_i \hat{p}_i$$

Here, $p = \frac{\phi\left(e^k g(x)\right) \odot y}{\left(\phi\left(e^k g(x)\right) \odot y\right) 1_l}$,

and $\hat{p} = 1_l^T - \frac{\phi(e^k(1-\hat{g}(x))) \odot y}{(\phi(e^k(1-\hat{g}(x))) \odot y) 1_l}$.

The overall loss function is:

$$\mathcal{L} = \mathcal{L}_{ori} + \mathcal{L}_{com} + \mu \mathcal{L}_{col}.$$

Experiment

Approaches	Data set							
	FG-NET	Lost	MSRCv2	Mirflickr	Soccer Player	Yahoo!News	FG-NET(MAE3)	FG-NET(MAE5)
PL-CL	0.072 ± 0.009	0.710 ± 0.022	0.469 ± 0.016	0.647 ± 0.012	0.534 ± 0.004	0.618 ± 0.003	0.433 ± 0.022	0.575 ± 0.015
PL-CL-PLCP	0.080 ± 0.009 •	0.763 ± 0.020 •	0.493 ± 0.013 •	0.665 ± 0.011 •	0.543 ± 0.002 •	0.625 ± 0.002 •	0.447 ± 0.017 •	0.595 ± 0.009 •
PL-AGGD	0.063 ± 0.010	0.690 ± 0.020	0.451 ± 0.023	0.610 ± 0.012	0.521 ± 0.004	0.605 ± 0.002	0.418 ± 0.020	0.562 ± 0.020
PL-AGGD-PLCP	0.076 ± 0.010 •	0.717 ± 0.020 •	0.473 ± 0.017 •	0.668 ± 0.014 •	0.534 ± 0.005 •	0.609 ± 0.002 •	0.441 ± 0.020 •	0.581 ± 0.014 •
SURE	0.052 ± 0.007	0.709 ± 0.022	0.445 ± 0.022	0.630 ± 0.022	0.519 ± 0.004	0.598 ± 0.002	0.356 ± 0.020	0.494 ± 0.021
SURE-PLCP	0.076 ± 0.011 •	0.719 ± 0.019 •	0.460 ± 0.020 •	0.657 ± 0.020 •	0.527 ± 0.004 •	0.606 ± 0.002 •	0.441 ± 0.019 •	0.582 ± 0.016 •
LALO	0.065 ± 0.010	0.682 ± 0.019	0.449 ± 0.016	0.629 ± 0.016	0.523 ± 0.003	0.601 ± 0.003	0.422 ± 0.023	0.566 ± 0.015
LALO-PLCP	0.076 ± 0.010 •	0.701 ± 0.019 •	0.453 ± 0.015 •	0.647 ± 0.018 •	0.529 ± 0.004 •	0.605 ± 0.002 •	0.443 ± 0.020 •	0.583 ± 0.014 •
PL-SVM	0.043 ± 0.008	0.406 ± 0.033	0.389 ± 0.029	0.516 ± 0.022	0.412 ± 0.006	0.509 ± 0.006	0.314 ± 0.019	0.445 ± 0.016
PL-SVM-PLCP	0.081 ± 0.011 •	0.688 ± 0.029 •	0.468 ± 0.025 •	0.607 ± 0.023 •	0.526 ± 0.005 •	0.609 ± 0.002 •	0.439 ± 0.021 •	0.583 ± 0.016 •
PL-KNN	0.036 ± 0.006	0.300 ± 0.018	0.393 ± 0.014	0.454 ± 0.016	0.492 ± 0.003	0.368 ± 0.004	0.288 ± 0.014	0.440 ± 0.017
PL-KNN-PLCP	0.076 ± 0.009 •	0.662 ± 0.025 •	0.469 ± 0.016 •	0.607 ± 0.023 •	0.523 ± 0.004 •	0.593 ± 0.004 •	0.434 ± 0.020 •	0.579 ± 0.016 •
	Average Improvement Ratio:	PL-CL: 3.61%	PL-AGGD: 5.10 %	SURE: 12.24 %	LALO: 4.01 %	PL-SVM: 39.26 %	PL-KNN: 53.98 %	

Table 6: Classification accuracy of each compared approach on the real-world data sets. For any compared approach $\mathcal{B}$, •/○ indicates whether $\mathcal{B}$-PLCP is statistically superior/inferior to $\mathcal{B}$ according to pairwise t-test at significance level of 0.05.

Approaches	CIFAR-10			CIFAR-100		
	$q = 0.1$	$q = 0.3$	$q = 0.5$	$q = 0.01$	$q = 0.05$	$q = 0.1$
PICO	94.39 ± 0.18 %	94.18 ± 0.12 %	93.58 ± 0.06 %	73.09 ± 0.34 %	72.74 ± 0.30 %	69.91 ± 0.24 %
PICO-PLCP	94.80 ± 0.07 % •	94.53 ± 0.10 % •	93.67 ± 0.16 % •	73.90 ± 0.20 % •	73.51 ± 0.21 % •	70.00 ± 0.35 %
Fully Supervised	$\mathcal{B}$: 94.91 ± 0.07 %		$\mathcal{B}$-PLCP: 95.02 ± 0.03 %	$\mathcal{B}$: 73.56 ± 0.10 %		$\mathcal{B}$-PLCP: 90.30 ± 0.08 %
PRODEN	89.12 ± 0.12 %	87.56 ± 0.15 %	84.92 ± 0.31 %	63.36 ± 0.33 %	60.88 ± 0.35 %	50.98 ± 0.74 %
PRODEN-PLCP	89.63 ± 0.15 % •	88.19 ± 0.19 % •	85.31 ± 0.31 % •	64.20 ± 0.25 % •	61.78 ± 0.29 •	50.76 ± 0.90 %
Fully Supervised	$\mathcal{B}$: 90.03 ± 0.13 %		$\mathcal{B}$-PLCP: 73.69 ± 0.14 %	$\mathcal{B}$: 65.03 ± 0.35 %		$\mathcal{B}$-PLCP: 65.52 ± 0.32 %

Table 2: Classification accuracy of each compared approach on CIFAR-10 and CIFAR-100. For any compared approach $\mathcal{B}$, •/○ indicates whether $\mathcal{B}$-PLCP is statistically superior/inferior to $\mathcal{B}$ according to pairwise t-test at significance level of 0.05.

Experiment

ParNeC 模式识别与神经计算研究组 PAttern Recognition and NEural Computing

Approaches	Data set							
	FG-NET	Lost	MSRCv2	Mirflickr	Soccer Player	Yahoo!News	FG-NET(MAE3)	FG-NET(MAE5)
PL-CL	0.159 ± 0.016	0.832 ± 0.019	0.585 ± 0.012	0.697 ± 0.019	0.715 ± 0.001	0.827 ± 0.003	0.600 ± 0.029	0.737 ± 0.018
PL-CL-PLCP	0.180 ± 0.011 •	0.852 ± 0.011 •	0.638 ± 0.008 •	0.704 ± 0.021 •	0.719 ± 0.002 •	0.829 ± 0.000 •	0.618 ± 0.023 •	0.748 ± 0.017 •
PL-AGGD	0.141 ± 0.012	0.793 ± 0.020	0.557 ± 0.015	0.695 ± 0.015	0.669 ± 0.003	0.808 ± 0.005	0.571 ± 0.027	0.709 ± 0.025
PL-AGGD-PLCP	0.165 ± 0.014 •	0.827 ± 0.019 •	0.640 ± 0.015 •	0.715 ± 0.015 •	0.713 ± 0.003 •	0.831 ± 0.004 •	0.599 ± 0.026 •	0.736 ± 0.019 •
SURE	0.158 ± 0.012	0.796 ± 0.026	0.603 ± 0.016	0.650 ± 0.024	0.700 ± 0.003	0.798 ± 0.005	0.590 ± 0.019	0.727 ± 0.020
SURE-PLCP	0.170 ± 0.013 •	0.834 ± 0.024 •	0.621 ± 0.013 •	0.699 ± 0.025 •	0.703 ± 0.003 •	0.827 ± 0.005 •	0.603 ± 0.024 •	0.738 ± 0.022 •
LALO	0.153 ± 0.017	0.818 ± 0.019	0.548 ± 0.009	0.681 ± 0.013	0.688 ± 0.004	0.822 ± 0.004	0.593 ± 0.025	0.730 ± 0.015
LALO-PLCP	0.168 ± 0.018 •	0.831 ± 0.019 •	0.620 ± 0.009 •	0.694 ± 0.019 •	0.706 ± 0.004 •	0.827 ± 0.004 •	0.604 ± 0.025 •	0.741 ± 0.019 •
PL-SVM	0.176 ± 0.015	0.609 ± 0.055	0.570 ± 0.040	0.581 ± 0.022	0.660 ± 0.008	0.691 ± 0.005	0.566 ± 0.025	0.706 ± 0.024
PL-SVM-PLCP	0.192 ± 0.012 •	0.786 ± 0.032 •	0.639 ± 0.031 •	0.628 ± 0.027 •	0.709 ± 0.006 •	0.821 ± 0.004 •	0.582 ± 0.025 •	0.719 ± 0.022 •
PL-KNN	0.041 ± 0.007	0.337 ± 0.030	0.415 ± 0.014	0.466 ± 0.013	0.493 ± 0.004	0.403 ± 0.010	0.285 ± 0.017	0.438 ± 0.015
PL-KNN-PLCP	0.166 ± 0.012 •	0.784 ± 0.031 •	0.635 ± 0.015 •	0.626 ± 0.019 •	0.698 ± 0.004 •	0.790 ± 0.008 •	0.576 ± 0.022 •	0.724 ± 0.019 •

Table 7: Transductive accuracy of each compared approach on the real-world data sets. For any compared approach $\mathcal{B}$, •/○ indicates whether $\mathcal{B}$-PLCP is statistically superior/inferior to $\mathcal{B}$ according to pairwise t-test at significance level of 0.05.

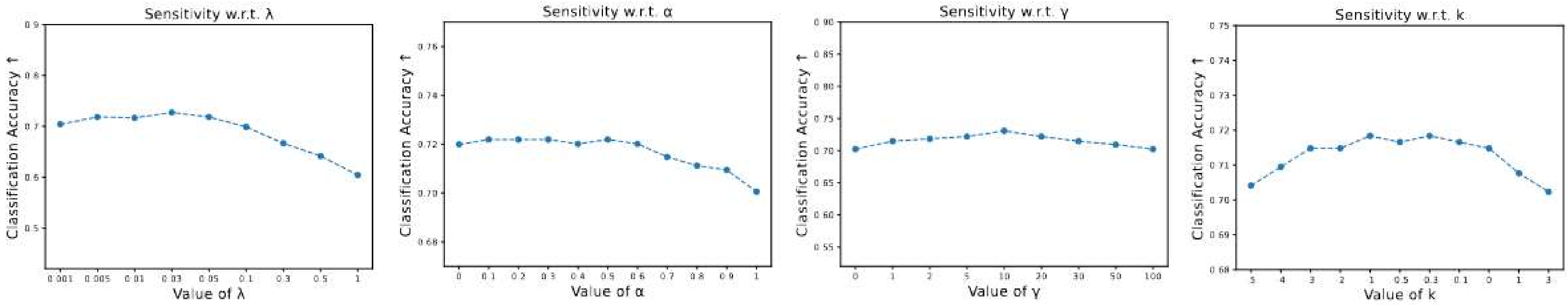
Experiment

ParNeC 模式识别与神经计算研究组 PAttern Recognition and NEural Computing

Kernel	Partner	Blur	Data set							
			FG-NET	Lost	MSRCv2	Mirflickr	Soccer Player	Yahoo!News	FG-NET(MAE3)	FG-NET(MAE5)
PL-AGGD			0.063 ± 0.010	0.690 ± 0.020	0.451 ± 0.023	0.610 ± 0.012	0.521 ± 0.004	0.605 ± 0.002	0.418 ± 0.020	0.562 ± 0.020
×	P	×	0.073 ± 0.011 ●	0.698 ± 0.023 ●	0.380 ± 0.013 ●	0.542 ± 0.013 ●	0.492 ± 0.003 ●	0.463 ± 0.002 ●	0.421 ± 0.020 ●	0.560 ± 0.016 ●
✓	P	×	0.073 ± 0.006 ●	0.721 ± 0.024 ○	0.471 ± 0.016 ●	0.664 ± 0.012 ●	0.521 ± 0.004 ●	0.608 ± 0.003 ●	0.422 ± 0.030 ●	0.566 ± 0.020 ●
✓	O	✓	0.071 ± 0.001 ●	0.721 ± 0.004 ○	$0.470 \pm 0,020$ ●	0.663 ± 0.011 ●	0.522 ± 0.003 ●	0.605 ± 0.002 ●	0.417 ± 0.022 ●	0.576 ± 0.014 ●
✓	P	✓	0.076 ± 0.010	0.717 ± 0.020	0.473 ± 0.017	0.668 ± 0.014	0.534 ± 0.005	0.609 ± 0.002	0.441 ± 0.020	0.581 ± 0.014

Table 8: Ablation study of PLCP coupled with PL-AGGD. ●/○ indicates whether PL-AGGD-PLCP is statistically superior/inferior to its degenerated version according to pairwise t-test at significance level of 0.05.

Experiment



(a) Sensitivity w.r.t. λ. (b) Sensitivity w.r.t. α. (c) Sensitivity w.r.t. γ. (d) Sensitivity w.r.t. k.

Figure 3: Sensitivity of PLCP.

Thanks