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

Abstract

In partial label learning (PLL), each instance is associated with a set of candidate labels among which only one is ground-truth. The majority of the existing works focuses on constructing robust classifiers to estimate the labeling confidence of candidate labels in order to identify the correct one. However, these methods usually struggle to rectify mislabeled samples. To help existing PLL methods identify and rectify mislabeled samples, in this paper, we introduce a novel partner classifier and propose a novel "mutual supervision" paradigm. Specifically, we instantiate the partner classifier predicated on the implicit fact that non-candidate labels of a sample should not be assigned to it, which is inherently accurate and has not been fully investigated in PLL. Furthermore, a novel collaborative term is formulated to link the base classifier and the partner one. During each stage of mutual supervision, both classifiers will blur each other's predictions through a blurring mechanism to prevent overconfidence in a specific label. Extensive experiments demonstrate that the performance and disambiguation ability of several well-established stand-alone and deep-learning based PLL approaches can be significantly improved by coupling with this learning paradigm.

Introduction

As a representative framework of weakly supervised learning, partial label learning (PLL), where each instance is associated with a set of candidate labels among which only one is ground-truth, has garnered increasing attention in recent years (Cour, Sapp, and Taskar 2011; Han et al. 2018; Jin and Ghahramani 2002a; Papandreou et al. 2015; Ren et al. 2018; Zhou 2018; Zhu and Goldberg 2009; Chai, Tsang, and Chen 2020; Ren et al. 2018; Li, Guo, and Zhou 2021; Li and Liang 2019). PLL has also been applied to many real-world applications due to the growing demand for identifying the valid label from a set of candidate labels. For instance, in the automatic face naming task (Zeng et al. 2013; Guillaumin, Verbeek, and Schmid 2010), each face crapped from images or videos is associated with a list of names extracted from the corresponding title or caption (Gong, Yuan, and Bao 2022). Another example is the facial age estimation task: for each human face, the ages annotated by crowd-sourcing labelers are considered as candidate labels (Panis et al. 2016).

*Corresponding author.

Formally speaking, suppose $\mathcal{X} = \mathbb{R}^q$ denotes the q-dimensional feature space and let $\mathcal{Y} = \{0, 1\}^l$ be the label space with l classes. Given a partial label data set $\mathcal{D} = \{\mathbf{x}_i, S_i | 1 \leq i \leq n\}$ where $\mathbf{x}_i \in \mathcal{X}$, $S_i \subseteq \mathcal{Y}$ is the corresponding candidate label set and n is the number of instances, the task of PLL is to induce a multi-class classifier $f : \mathcal{X} \to \mathcal{Y}$ based on $\mathcal{D}$. The most challenging part of PLL is that the ground-truth label y_i of a training sample $\mathbf{x}_i$ conceals in its candidate labels, i.e, $y_i \in S_i$, which cannot be directly accessible during the training process.

To address this challenge, existing works focus on disambiguation, which involves differentiating the labeling confidences of each candidate label to identify the ground truth. This process typically relies on an alternative and iterative method for updating the classifier's parameters. For instance, PL-AGGD (Wang, Zhang, and Li 2022) constructs a similarity graph to achieve disambiguation, and SURE (Feng and An 2019) aims to maximize an infinity norm to differentiate the ground-truth label. 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? To explore this question, we conduct experiments on a real-world data set Lost (Cour et al. 2009) and record the labeling confidences of several candidate labels generated by PL-AGGD (Wang, Zhang, and Li 2022) in each iteration, which is shown in Fig. 1. Our findings reveal some intriguing phenomena:

- Each candidate label's labeling confidence is likely to continually increase or decrease until convergence.
- For a false positive candidate label with a large labeling confidence, although its confidence may decrease in subsequent iterations, the confidence remains substantial and can easily lead to the incorrect identification of the ground truth label.

The observed phenomena suggest that once the labeling confidence of a false positive candidate label increases, it becomes difficult to decrease in the subsequent iterations. Furthermore, even if the confidence of a false positive candidate label decreases appropriately, it may still be recognized as the ground truth one, as its initial labeling confidence remains large and continues to be greater than the confidence of the ground truth label upon convergence. As a result, cor-

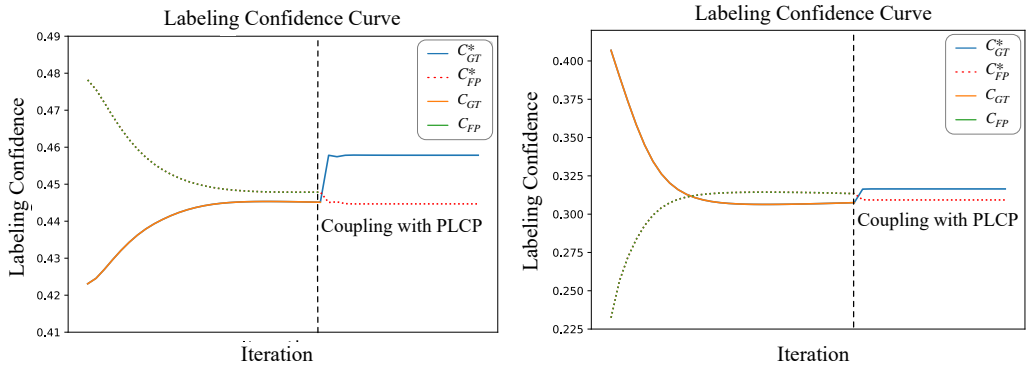

Figure 1: Two representative errors a typical PLL classifier, e.g., PL-AGGD (Wang, Zhang, and Li 2022) may make. C_{GT} and C^*_{GT} stand for the labeling confidence of the ground-truth label generated by PL-AGGD and PL-AGGD coupled with PLCP, and C_{FP} and C^*_{FP} stand for that of a false positive label predicted by PL-AGGD and PL-AGGD coupled with PLCP. (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). The labeling confidence of a false positive candidate label keeps increasing and becomes the largest, which misleads the final prediction. When coupled with PLCP (vertical dashed line), the new labeling confidence of each candidate label generated by the partner classifier is adopted as the supervision to help PL-AGGD correct these errors, which results in a mutation in the figures.

recting mislabeled samples for a PLL classifier itself proves to be quite challenging.

To address the aforementioned challenge, we propose a novel PLL mutual supervision framework called PLCP, referring to **P**artial **L**abel Learning with a **C**lassifier as **P**artner. Given a classifier provided by any existing PLL approach as the base classifier, as shown in Fig. 2, PLCP introduces an additional partner classifier to help the base classifier identify and rectify mislabeled samples. This partner classifier aims to provide more accurate and complementary information to the base classifier, thereby enabling better disambiguation and facilitating mutual supervision between the two classifiers. It is worth noting that the design of an effective partner classifier is crucial to the success of this framework, as the feedback from the partner classifier greatly influences the base classifier's ability to identify and correct mislabeled samples. Since the information in non-candidate labels, which indicates that a set of labels DO NOT belong to a sample, is more precise than that in candidate labels and is often overlooked by the majority of existing works, the partner classifier can be designed to specify the labels that should not be assigned to a sample, thereby complementing the base classifier. Additionally, a novel collaborative term is also designed to link the base classifier and the partner classifier.

In each stage of mutual supervision, the labeling confidence is first updated based on the base classifier's modeling output. Subsequently, a blurring mechanism is employed to further process the labeling confidence to introduce uncertainty, which could potentially reduce the large confidence of some false positive candidate labels or increase the small confidence of the ground truth one. This updated labeling confidence then serves as the supervision information to interact with the partner classifier, whose final output will also be converted to supervise the base classifier. The predictions of the two classifiers, while distinct, are inextricably linked, enhancing the disambiguation ability of this paradigm in two opposing ways. With this mutual supervision paradigm, mislabeled instances have a higher likelihood of being corrected.

The main contributions of this paper can be summarized as follows:

- We highlight two representative errors that a PLL classifier may make, which has not been previously investigated and offers a new insight into PLL.
- We introduce a partner classifier based on non-candidate labels to better identify and correct mislabeled samples of a base classifier through a mutual supervision framework, which is applicable to all types of PLL approaches.
- We propose a novel collaborative term in the partner classifier, which links the base classifier and itself. Additionally, a blurring mechanism is introduced to add uncertainty to the outputs, which effectively tackles the mentioned drawbacks.
- We conduct experiments on several data sets to validate the effectiveness of this framework, and the results demonstrate that PLCP improves the disambiguation ability of the base classifier, leading to outstanding performance across all data sets.

Related Work

Partial label learning (PLL), also known as superset-label learning (Liu and Dietterich 2012, 2014) or ambiguous label learning (Hüllermeier and Beringer 2005; Zeng et al. 2013), 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

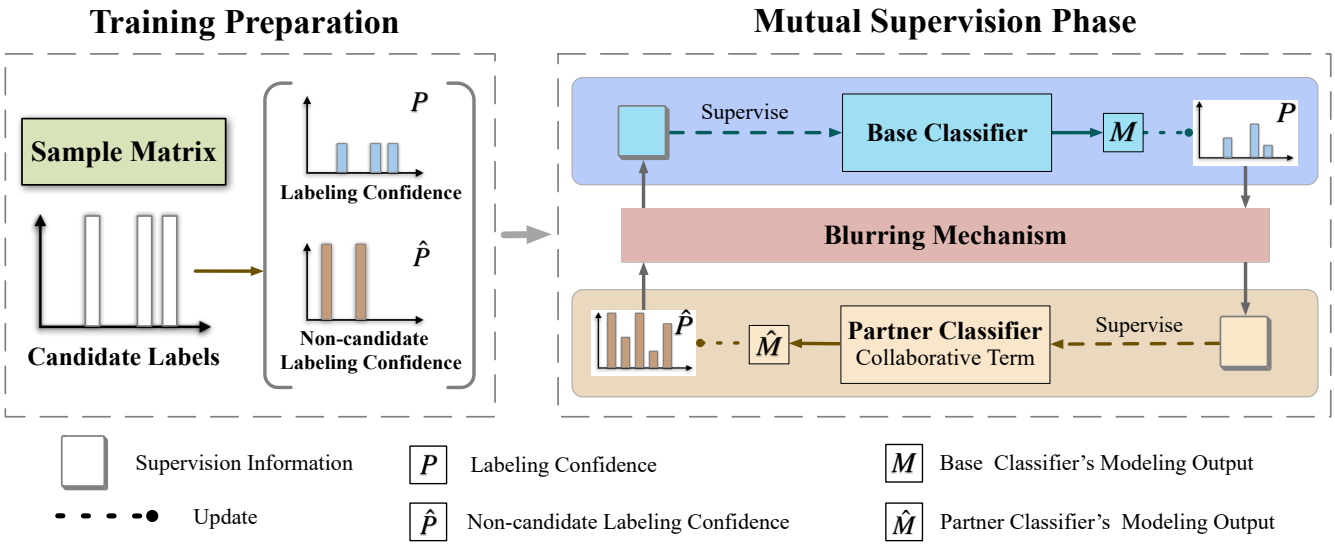

Figure 2: The framework of PLCP. A partner classifier is constructed based on the non-candidate label information to enable mutual supervision between the base classifier and itself. In each stage of mutual supervision, the base classifier updates the labeling confidence $\mathbf{P}$ based on its modeling output $\mathbf{M}$ and blurs it through a blurring mechanism. Afterwards, the output is represented as the supervision information to interact with the partner classifier. The pipeline of the partner classifier is almost the same as the base classifier's.

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.

To tackle the mentioned challenge, existing works mainly focus on disambiguation (Feng and An 2019; Nguyen and Caruana 2008; Zhang and Yu 2015; Wang, Zhang, and Li 2022; Fan et al. 2021; Xu, Lv, and Geng 2019; Zhang, Wu, and Bao 2022; Qian et al. 2023), which can be broadly divided into two categories: averaging-based approaches and identification-based approaches. For the averaging-based approaches (Hüllermeier and Beringer 2005; Cour, Sapp, and Taskar 2011; Zhang and Yu 2015), each candidate label of a training sample is treated equally as the ground-truth one and the final prediction is yielded by averaging the modeling outputs. For instance, PL-KNN (Hüllermeier and Beringer 2005) averages the candidate labels of neighboring samples to make the prediction. This kind of approach is intuitive, however, it can be easily influenced by false positive candidate labels which results in inferior performance. For identification-based approaches, (Feng and An 2018, 2019; Nguyen and Caruana 2008; Jin and Ghahramani 2002b; Yu and Zhang 2017), the ground-truth label is treated as a latent variable and can be identified through an iterative optimization procedure such as EM. Moreover, labeling confidence based strategy is proposed in many state-of-the-art identification based approaches for better disambiguation. (Zhang and Yu 2015) and (Wang, Zhang, and Li 2022) construct a similarity graph based on the feature space to generate labeling confidence of candidate labels.

Recently, deep-learning based models have been introduced to address PLL tasks (Lv et al. 2020; Xu et al. 2021; He et al. 2022; Wu, Wang, and Zhang 2022; Lyu, Wu, and Feng 2022; Xia et al. 2023). PICO (Wang et al. 2022) is a contrastive learning-based approach devised to tackle label ambiguity in partial label learning. This method seeks to discern the ground-truth label from the candidate set by utilizing contrastively learned embedding prototypes. Lv et al. proposed PRODEN in (Lv et al. 2020), a model where the simultaneous updating of the model and identification of true labels are seamlessly integrated. Furthermore, He et al. introduced a partial label learning method based on semantic label representations in (He et al. 2022). This method employs a novel weighted calibration rank loss to facilitate label disambiguation. By leveraging label confidence, the approach weights the similarity towards all candidate labels and subsequently yields a higher similarity of candidate labels in comparison to each non-candidate label.

However, as mentioned in Section 1, the above kinds of approach usually fail to identify and correct the mislabeled samples. To address this challenge, we propose a novel framework called PLCP in the next section.

The Proposed Approach

Denote $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_n]^\top \in \mathbb{R}^{n \times q}$ the sample matrix with n instances, and $\mathbf{Y} = [\mathbf{y}_1, \mathbf{y_2}, ..., \mathbf{y}_n]^\top \in \{0, 1\}^{n \times l}$ the partial label matrix with l labels, where $y_{ij} = 1$ (resp. $y_{ij} = 0$) if the j-th label of $\mathbf{x}_i$ resides in (resp. does not reside in) its candidate label set. Given the partial label data set $\mathcal{D} = \{\mathbf{x}_i, S_i | 1 \leq i \leq n\}$, the task of PLL is to learn a multi-class classifier $f : \mathcal{X} \to \mathcal{Y}$ based on $\mathcal{D}$.

To address the challenge described in Section 1, a partner classifier $\mathcal{P}$ is designed to complement a base classifier $\mathcal{B}$ and also can be supervised by $\mathcal{B}$. $\mathcal{B}$ represents any existing PLL

classifier with fine generality and flexibility. In each stage of mutual supervision, the base classifier updates the labeling confidence based on its modeling outputs, and then a blurring mechanism is applied to further process it. Subsequently the output of the labeling confidence is taken as the supervision information for the partner classifier. The learning pipeline of the partner classifier closely mirrors that of the base classifier. The following subsections will provide further details on this process.

Base Classifier

Suppose $\mathbf{P} = [\mathbf{p}_1, \mathbf{p}_2, ..., \mathbf{p}_n]^\mathsf{T} \in \mathbb{R}^{n\times l}$ is the labeling confidence matrix, where $\mathbf{p}_i \in \mathbb{R}^l$ represents the labeling confidence vector of $\mathbf{x}_i$, and p_{ij} denotes the probability of the j-th label being the ground-truth label of $\mathbf{x}_i$. For the base classifier utilizing the labeling confidence strategy, $\mathbf{P}$ is initialized according to the base classifier. Otherwise, it is initialized as follows:

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

The labeling confidence vector $\mathbf{p}_i$ typically satisfies the following constraints: $\sum_j p_{ij} = 1$, $0 \le p_{ij} \le y_{ij}$ (Wang, Zhang, and Li 2022; Feng and An 2018). The first constrains $\mathbf{p}_i$ to be normalized, and the second indicates that only the confidence of a candidate label has a chance to be positive. It should be noted that the ideal state of $\mathbf{p}_i$ is one-hot.

Once the base classifier has been trained, its modeling output will be generated. Denote the modeling output matrix $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, ..., \mathbf{m}_n]^\mathsf{T} \in \mathbb{R}^{n\times l}$, where $\mathbf{m}_{ij}$ represents the probability of the j-th label being $\mathbf{x}_i$'s ground-truth label as predicted by the base classifier. It should be noted that $\mathbf{m}_i$ may either be a real-valued or one-hot vector depending on the specific base classifier. For instance, for SURE (Feng and An 2019) $\mathbf{m}_i$ is real-valued, while for PL-KNN (Hüllermeier and Beringer 2005) it is one-hot. Afterwards, $\mathbf{P}$ is updated to $\mathbf{P}_1$ through the following equation:

$$\mathbf{P}_1 = \mathcal{T}_0\left(\mathcal{T}_\mathbf{Y}\left(\alpha\mathbf{P} + (1-\alpha)\mathbf{M}\right)\right), \tag{2}$$

where α is a hyper-parameter controlling the smoothness of the labeling confidence. $\mathcal{T}_0$, $\mathcal{T}_\mathbf{Y}$ are two thresholding operators in element-wise, i.e., $\mathcal{T}_0(a) := \max\{0, a\}$ with a being a scalar and $\mathcal{T}_\mathbf{Y}(a) := \min\{y_{ij}, a\}$.

Blurring Mechanism

In the next step, a blurring mechanism is designed to further process the labeling confidence matrix $\mathbf{P}_1$ to $\mathbf{Q}_1$:

$$\mathbf{Q}_1 = \phi(e^k\mathbf{P}_1) \odot \mathbf{Y}, \tag{3}$$

where $\phi(\cdot)$ is an element-wise operator, for a matrix $\mathbf{A} = [a_{ij}]_{n\times l} \in \mathbb{R}^{n\times l}$, $\phi(\mathbf{A}) = [\exp(a_{ij})]_{n\times l}$. k is a temperature parameter that controls the extent of blurring of labeling confidence. The labeling confidences are expected to be blurred at each stage of mutual supervision to prevent being overconfident in some false positive labels, hence we set $k < 0$, which means two labeling confidences that differ significantly can also become close. $\odot$ represents the Hadamard product of two matrices. For matrices $\mathbf{A}$ and $\mathbf{B}$ with same size $n \times l$, $\mathbf{A} \odot \mathbf{B} = [a_{ij}b_{ij}]_{n\times l}$. The Hadamard product allows only the candidate label has a positive confidence. We then normalize each row of $\mathbf{Q}_1$ to satisfy the two constraints of labeling confidence, and output the result $\mathbf{O}_1 \in \mathbb{R}^{n\times l}$.

Partner Classifier

Since the partner classifier has significant impacts on the success of PLCP, designing an appropriate partner classifier is quite important. In order to better assist the base classifier, a classifier that specifies the labels that should not be assigned to a sample is instantiated as the partner classifier, since non-candidate label information is exactly accurate and opposite to the candidate label information, making it a valuable complement to the base classifier.

Denote the non-candidate label matrix $\hat{\mathbf{Y}} = [\hat{y}_{ij}]_{n\times l}$ 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 $\mathbf{x}_i$, and $\hat{\mathbf{P}} = [\hat{\mathbf{p}}_1, \hat{\mathbf{p}}_2, ..., \hat{\mathbf{p}}_n]^\mathsf{T} \in \mathbb{R}^{n\times l}$ the non-candidate labeling confidence matrix, where $\hat{p}_{ij}$ represents the probability of the j-th label NOT being the ground-truth label of $\mathbf{x}_i$. Similar to the labeling confidence, $\hat{\mathbf{P}}$ is also constrained with two constraints: $\sum_j \hat{p}_{ij} = l - 1, \hat{y}_{ij} \le \hat{p}_{ij} \le 1$. The first term constrains the sum of the probability of each label being invalid is strictly $l - 1$, and the second indicates that only candidate label has a chance to update its non-candidate labeling confidence while the others keep 1 (invalid labels). Note that the ideal state of $\hat{\mathbf{p}}_i$ is "zero-hot", i.e., only one element in $\hat{\mathbf{p}}_i$ is 0 with others 1. $\hat{\mathbf{P}}$ is initialized as $\hat{\mathbf{Y}}$. Suppose $\hat{\mathbf{W}} = [\hat{\mathbf{w}}_1, \hat{\mathbf{w}}_2, ..., \hat{\mathbf{w}}_q]^\mathsf{T} \in \mathbb{R}^{q\times l}$ is the weight matrix, the partner classifier is formulated as follows:

$$\begin{aligned} \min_{\hat{\mathbf{W}},\hat{\mathbf{b}},\mathbf{C}} \quad & \left\|\mathbf{X}\hat{\mathbf{W}} + \mathbf{1}_n\hat{\mathbf{b}}^\mathsf{T} - \mathbf{C}\right\|_F^2 + \lambda\left\|\hat{\mathbf{W}}\right\|_F^2 \\ \text{s.t.} \quad & \hat{\mathbf{Y}} \le \mathbf{C} \le \mathbf{1}_{n\times l}, \mathbf{C}\mathbf{1}_l = (l-1)\mathbf{1}_n, \end{aligned} \tag{4}$$

where $\hat{\mathbf{b}} = [\hat{b}_1, \hat{b}_2, ..., \hat{b}_l]^\mathsf{T} \in \mathbb{R}^l$ is the bias term, $\mathbf{1}_n \in \mathbb{R}^n$ is an all one vectors, $\mathbf{1}_{n\times l}$ is an all one matrix with size $n \times l$ and λ is a hyper-parameter trading off these terms. $\|\hat{\mathbf{W}}\|_F$ is the Frobenius norm of the weight matrix. $\mathbf{C} \in \mathbb{R}^{n\times l}$ represents non-candidate labeling confidence, which is a temporary variable only used for optimization in the partner classifier. By solving this optimization problem, the partner classifier learns to specify which labels should not be assigned to each sample, improving the overall performance of the mutual supervision process.

The Collaborative Term

Since a label is either ground-truth or not and based on the constraints on $\mathbf{p}_i$ and $\hat{\mathbf{p}}_i$, the smallest value of $\mathbf{p}_i^\mathsf{T}\hat{\mathbf{p}}_i$ is obtained when $\hat{\mathbf{p}}_i$ is zero-hot ($\mathbf{p}_i$ is one-hot). Therefore, a collaborative relationship between the outputs of the partner classifier and the prior given by the base classifier can be formulated, with the partner classifier becoming

$$\begin{aligned} \min_{\hat{\mathbf{W}},\hat{\mathbf{b}},\mathbf{C}} \quad & \left\|\mathbf{X}\hat{\mathbf{W}} + \mathbf{1}_n\hat{\mathbf{b}}^\mathsf{T} - \mathbf{C}\right\|_F^2 + \gamma\mathrm{tr}\left(\mathbf{O}_1\mathbf{C}^\mathsf{T}\right) + \lambda\left\|\hat{\mathbf{W}}\right\|_F^2 \\ \text{s.t.} \quad & \hat{\mathbf{Y}} \le \mathbf{C} \le \mathbf{1}_{n\times l}, \mathbf{C}\mathbf{1}_l = (l-1)\mathbf{1}_n, \end{aligned} \tag{5}$$

where γ is a hyper-parameter. Different from candidate labels, the non-candidate labels are directly accessible and accurate,

therefore the information learnt by the partner classifier is precise, which can effectively complement the base classifier.

The problem in Eq. (5) can be solved via an alternative and iterative manner, which results in an optimal weight matrix $\hat{\mathbf{W}}$ and bias term $\hat{\mathbf{b}}$. For detailed solution process, please refer to the Supplementary. The modeling output $\hat{\mathbf{M}}$ for the training data is

$$\hat{\mathbf{M}} = \mathbf{X}\hat{\mathbf{W}} + \mathbf{1}_n\hat{\mathbf{b}}^{\mathsf{T}}. \tag{6}$$

Afterwards, the non-candidate labeling confidence $\hat{\mathbf{P}}$ is updated to $\hat{\mathbf{P}}_1$ following

$$\hat{\mathbf{P}}_1 = \mathcal{T}_1\left(\mathcal{T}_{\hat{\mathbf{Y}}}\left(\alpha\hat{\mathbf{P}} + (1-\alpha)\hat{\mathbf{M}}\right)\right), \tag{7}$$

where $\mathcal{T}_1$, $\mathcal{T}_{\hat{\mathbf{Y}}}$ are two thresholding operators in element-wise, i.e., $\mathcal{T}_1(m) := \min\{1, m\}$ with m being a scalar and $\mathcal{T}_{\hat{\mathbf{Y}}}(m) := \max\{\hat{y}_{ij}, m\}$. The non-candidate labeling confidence can be further processed as $\hat{\mathbf{Q}}_1$ via the blurring mechanism:

$$\hat{\mathbf{Q}}_1 = \phi(e^k(1-\hat{\mathbf{P}}_1)) \odot \mathbf{Y}. \tag{8}$$

For convenience, we transform the non-candidate labeling confidence into labeling confidence in advance, and finally $\hat{\mathbf{Q}}_1$ is normalized as $\hat{\mathbf{O}}_1$, which is the supervision of the base classifier in the next iteration. For an unseen sample $\mathbf{x}$, suppose the non-candidate labeling confidence vector predicted by the partner classifier in the last iteration is $\hat{\mathbf{p}}^{pt}$. The prediction y^* of PLCP is

$$y^* = \operatorname{argmax}_i \quad (1-\hat{p}_i^{pt}), \tag{9}$$

Extensions of PLCP

We can also extend PLCP to a kernel version which extends the feature map to a higher dimensional space or a deep-learning based version which enables deep-learning based methods involved in PLCP. For more details, please refer to the Supplementary.

Experiments

Compared Approaches

To evaluate the effectiveness of PLCP, we couple it with several well-established partial label learning approaches. Suppose $\mathcal{B}$ represents any partial label learning classifier (i.e., the base classifier) and $\mathcal{B}$-PLCP is the $\mathcal{B}$ coupled with PLCP, the performances of $\mathcal{B}$ and $\mathcal{B}$-PLCP are compared to verify the effectiveness of PLCP. In this paper, $\mathcal{B}$ is instantiated by six stand-alone (non-deep) approaches, PL-CL(Jia, Si, and Zhang 2023), PL-AGGD (Wang, Zhang, and Li 2022), SURE (Feng and An 2019), LALO (Feng and An 2018), PL-SVM (Nguyen and Caruana 2008) and PL-KNN (Hüllermeier and Beringer 2005), and two deep-learning based methods PICO (Wang et al. 2022) and PRODEN (Lv et al. 2020). The hyper-parameters of $\mathcal{B}$ are all set according to the original papers.

Comparison with Stand-alone Methods

Experimental Settings For PLCP, we set $\lambda = 0.05$, $\alpha = 0.5$, $\gamma = 2$ and $k = -1$, and the maximum iteration of mutual supervision is set to 5. For PL-CL, PL-AGGD, SURE, LALO and PL-SVM, the kernel function is Gaussian function, which is the same as we adopt. Ten runs of 50%/50% random train/test splits are performed, and the average accuracy with standard deviation is represented for all $\mathcal{B}$ and $\mathcal{B}$-PLCP. For PL-SVM and PL-KNN which do not adopt labeling confidence strategy, $\hat{\mathbf{O}}$ is further processed as $\mathcal{G}(\hat{\mathbf{O}} - \mathbf{P})$ where $\mathcal{G}$ is an element-wise operator and $\mathcal{G}(x) = 1$ if $x \geq 0$ otherwise 0 with x being a scalar.

We conduct experiments on six real-world partial label data sets collected from several domains and tasks, including FG-NET (Panis et al. 2016) for facial age estimation, Lost (Cour et al. 2009), Soccer Player (Zeng et al. 2013) and Yahoo!News (Guillaumin, Verbeek, and Schmid 2010) for automatic face naming, MSRCv2 (Liu and Dietterich 2012) for object classification and Mirflickr (Huiskes and Lew 2008) for web image classification. The details of the data sets are summarized in the Supplementary.

For facial age estimation task, the ages annotated by crowdsourced labelers are considered as each human face's candidate labels. For automatic face naming task, each face scratched from a video or an image is presented as a sample while the names extracted from the corresponding titles or captions are its candidate labels. For object classification task, image segmentations are taken as instances with objects appearing in the same image as candidate labels.

Performance on Real-World Data Set It is noteworthy that the number of average label of FG-NET is quite large, which could cause quite low classification accuracy for all approaches. The common strategy is to evaluate the mean absolute error (MAE) between the predicted age and the ground-truth one. Specifically, we add another two sets of comparisons on FG-NET w.r.t. MAE3/MAE5, which means that test samples can be considered to be correctly classified if the difference between the predicted age and true age is no more than 3/5 years. The results of these two comparisons are shown in the Supplementary due to the page limit. Table 1 summaries the classification accuracy with standard deviation of each approach on real-world data sets, where we can observe that

- $\mathcal{B}$-PLCP significantly outperforms the base classifier $\mathcal{B}$ in all cases according to the pairwise t-test with a significance level of 0.05, which validates the effectiveness of PLCP.
- State-of-the-art (SOTA) approaches, such as PL-CL, PL-AGGD, LALO and SURE, can also be significantly improved by PLCP on all data sets. For instance, on FG-NET the performance of SURE can be improved by **45%**, and on FG-NET(MAE3) the performance of LALO can be improved by **5%**. Additionally, PL-AGGD can be improved by **5.10%** and PL-CL can be improved by **3.61%**. The partner classifier's non-candidate label information effectively and significantly aids disambiguation, leading to outstanding performance of the PLCP framework.
- The performances of PL-SVM and PL-KNN are improved significantly and impressively when coupled with PLCP. For example, on FG-NET the performance of PL-KNN-PLCP is **more than two times** better than that of PL-KNN, and PL-SVM-PLCP's performance is **more than**

Approaches	Data set					
	FG-NET	Lost	MSRCv2	Mirflickr	Soccer Player	Yahoo!News
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
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 •
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
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 •
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
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 •
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
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 •
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
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 •
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
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 •
Improvement:	PL-CL: 3.61%	PL-AGGD: 5.10 %	SURE: 12.24 %	LALO: 4.01 %	PL-SVM: 39.26 %	PL-KNN: 53.98 %

Table 1: 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:	73.69 ± 0.14 %
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: 90.30 ± 0.08 %		$\mathcal{B}$: 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.

Approaches	Data set					
	FG-NET	Lost	MSRCv2	Mirflickr	Soccer Player	Yahoo!News
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 =
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 •
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
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 •
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
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 •
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
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 •
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
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 •
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
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 •

Table 3: 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.

Kernel	Partner	Blur	Data set					
			FG-NET	Lost	MSRCv2	Mirflickr	Soccer Player	Yahoo!News
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
×	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 •
✓	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 •
✓	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 •
✓	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

Table 4: 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.

1.5 times better than that of PL-SVM on Lost. Although PL-SVM and PL-KNN are inferior to the SOTA methods, **they can achieve almost the same performance as SOTA's when coupled with PLCP**.

Comparison with Deep-learning Based Methods

Experimental Settings We conduct experiments on two benchmarks CIFAR-10 and CIFAR-100 (Krizhevsky, Hinton et al. 2009), and following the settings in (Wang et al. 2022; Lv et al. 2020), we generate false positive labels by flipping negative labels $\hat{y} \neq y$ for each sample with a probability $q = P(\hat{y} \in \mathcal{Y} | \hat{y} \neq y)$, where y is the ground-truth label. In other words, the probability of a false positive is uniform across all $l-1$ negative labels. We combine the flipped ones with the ground-truth label to create the set of candidate labels. Specifically, q is set to 0.1, 0.3 and 0.5 for CIFAR-10 and 0.01, 0.05 and 0.1 for CIFAR-100. Five independent runs are performed with the the average accuracy and standard deviation recorded (with different seeds).

Performance on CIFAR-10 and CIFAR-100 Table 2 presents the classification accuracy with standard deviation of each approach on CIFAR-10 and CIFAR-100, where we can find that

- $\mathcal{B}$-PLCP consistently outperforms the base classifier $\mathcal{B}$ in 83.3% cases, which confirms that PLCP effectively improves the performance of deep-learning based models. Notably, $\mathcal{B}$-PLCP is never significantly outperformed by any $\mathcal{B}$.
- Although the performance improvement of $\mathcal{B}$-PLCP on different settings appears to be limited, it is important to note that its performance is very close to that of fully supervised $\mathcal{B}$. In some cases, it even surpasses the performance of fully supervised one.

Further Analysis

Improvement of the Disambiguation Ability Transductive accuracy (i.e., classification accuracy on training samples) reflects the disambiguation ability of a PLL approach (Wang, Zhang, and Li 2022; Cour, Sapp, and Taskar 2011; Zhang, Zhou, and Liu 2016). In order to validate whether PLCP can correct some mislabeled samples and truly improve the disambiguation ability of $\mathcal{B}$, we summarize the transductive accuracy of $\mathcal{B}$ and $\mathcal{B}$-PLCP on real-world data sets and the results are shown in Table 3. It is obvious that $\mathcal{B}$-PLCP outperforms $\mathcal{B}$ in all cases according to the pairwise t-test with a significance level of 0.05, which validates that the disambiguation ability of $\mathcal{B}$ can be truly improved by PLCP. In other words, by integrating PLCP, classifiers can better identify and correct mislabeled samples, leading to outstanding performance.

Effectiveness of the Non-candidate Label Information In order to validate the effectiveness of the non-candidate label information, we instantiate the partner classifier with the form as

$$\begin{aligned} \min_{\mathbf{W},\mathbf{b},\mathbf{L}} \quad & \left\|\mathbf{X}\mathbf{W} + \mathbf{1}_n\mathbf{b}^\mathsf{T} - \mathbf{L}\right\|_F^2 + \lambda \left\|\hat{\mathbf{W}}\right\|_F^2 \\ & + \gamma \mathrm{tr}\left(\mathbf{O}_1(\mathbf{1}_{n\times l} - \mathbf{L})^\mathsf{T}\right) \\ \text{s.t.} \quad & \mathbf{0}_{n\times l} \leq \mathbf{L} \leq \mathbf{Y}, \mathbf{L}\mathbf{1}_l = \mathbf{1}_n, \end{aligned} \tag{10}$$

where $\mathbf{W}$ and $\mathbf{b}$ are two model parameters and $\mathbf{L} \in \mathbb{R}^{n\times l}$ is the labeling confidence matrix. Denote this classifier O and the original partner classifier P, the classification accuracy of PL-AGGD mutual-supervised with O on each real-world data set is shown in the third row of Table 4. It is obvious that the performance of PL-AGGD coupled with O is significantly inferior to that with P in 87.5% cases, which validates the usefulness of the non-candidate label information. The accurate non-candidate label information provided by the partner classifier effectively complements the base classifier, leading to better performance.

Usefulness of the Blurring Mechanism It is also interesting to investigate the usefulness of the blurring mechanism in PLCP. By simply normalizing $\mathbf{P}_1$ to $\mathbf{O}_1$ and $\hat{\mathbf{P}}_1$ to $\hat{\mathbf{O}}_1$ (i.e., skipping Eq. (3) and Eq. (8)), the classification accuracy of PLCP without this mechanism is recorded in the second row of Table 4, where we can find that the performance of PL-AGGD-PLCP w/o Blur is significantly inferior to those with Blur in 87.5% cases. The blurring mechanism effectively tackles the overconfidence issue in PLL, leading to outstanding performance.

Furthermore, we examine the impact of various values of k on the performance of PLCP, as depicted in Fig. 1(d) in the Supplementary. It can be clearly observed that when k is too small, the performance of PLCP deteriorates significantly. Additionally, the performance will be also exacerbated when $k \geq 0$, for in this case the predictions of the classifiers are enhanced rather than being blurred, amplifying small differences between two values. In this case, PLCP contributes little to disambiguation, resulting in inferior performance.

Influence of Kernel Extension We also conduct experiments to show the improvement of kernel extension used in partner classifier. Comparing the results in the first and second rows in Table 4, we can observe that the performance of PL-AGGD coupled with the classifier using kernel extension is superior to that without kernel extension on all the data sets, which validates the effectiveness of the kernel.

Additionally, we also conduct experiments on sensitivity of different hyper-parameters in the Supplementary.

Conclusion

In this paper, a novel mutual supervision paradigm in partial label learning called PLCP is proposed. Specifically, a partner classifier is introduced and a novel collaborative term is designed to link the base classifier and the partner classifier, which enables mutual supervision between the two classifiers. A blurring mechanism is involved in this paradigm for better disambiguation. Comprehensive experiments validate the outstanding performance of PLCP coupling with stand-alone approaches and deep-learning based methods, which further validates that the mislabeled examples can be identified and corrected by PLCP. In the future, it is also interesting to investigate other methods to identify and correct mislabeled samples.

Acknowledgements

This work was supported by NSFC 62176159, 62322604, Natural Science Foundation of Shanghai 21ZR1432200, and Shanghai Municipal Science and Technology Major Project 2021SHZDZX0102.

References

Chai, J.; Tsang, I. W.; and Chen, W. 2020. Large Margin Partial Label Machine. *IEEE Trans. Neural Networks Learn. Syst.*, 31(7): 2594–2608.

Cour, T.; Sapp, B.; Jordan, C.; and Taskar, B. 2009. Learning from ambiguously labeled images. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, 919–926. IEEE Computer Society.

Cour, T.; Sapp, B.; and Taskar, B. 2011. Learning from Partial Labels. *J. Mach. Learn. Res.*, 12: 1501–1536.

Fan, J.; Yu, Y.; Wang, Z.; and Gu, J. 2021. Partial label learning based on disambiguation correction net with graph representation. *IEEE Transactions on Circuits and Systems for Video Technology*, 32(8): 4953–4967.

Feng, L.; and An, B. 2018. Leveraging Latent Label Distributions for Partial Label Learning. In Lang, J., ed., *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden*, 2107–2113. ijcai.org.

Feng, L.; and An, B. 2019. Partial Label Learning with Self-Guided Retraining. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 3542–3549. AAAI Press.

Gong, X.; Yuan, D.; and Bao, W. 2022. Partial Label Learning via Label Influence Function. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvári, C.; Niu, G.; and Sabato, S., eds., *International Conference on Machine Learning, ICML 2022, 17-23 July 2022, Baltimore, Maryland, USA*, volume 162 of *Proceedings of Machine Learning Research*, 7665–7678. PMLR.

Guillaumin, M.; Verbeek, J.; and Schmid, C. 2010. Multiple Instance Metric Learning from Automatically Labeled Bags of Faces. In Daniilidis, K.; Maragos, P.; and Paragios, N., eds., *Computer Vision - ECCV 2010, 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I*, volume 6311 of *Lecture Notes in Computer Science*, 634–647. Springer.

Han, B.; Yao, Q.; Yu, X.; Niu, G.; Xu, M.; Hu, W.; Tsang, I. W.; and Sugiyama, M. 2018. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 8536–8546.

He, S.; Feng, L.; Lv, F.; Li, W.; and Yang, G. 2022. Partial Label Learning with Semantic Label Representations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 545–553.

Huiskes, M. J.; and Lew, M. S. 2008. The MIR flickr retrieval evaluation. In Lew, M. S.; Bimbo, A. D.; and Bakker, E. M., eds., *Proceedings of the 1st ACM SIGMM International Conference on Multimedia Information Retrieval, MIR 2008, Vancouver, British Columbia, Canada, October 30-31, 2008*, 39–43. ACM.

Hüllermeier, E.; and Beringer, J. 2005. Learning from Ambiguously Labeled Examples. In Famili, A. F.; Kok, J. N.; Sánchez, J. M. P.; Siebes, A.; and Feelders, A. J., eds., *Advances in Intelligent Data Analysis VI, 6th International Symposium on Intelligent Data Analysis, IDA 2005, Madrid, Spain, September 8-10, 2005, Proceedings*, volume 3646 of *Lecture Notes in Computer Science*, 168–179. Springer.

Jia, Y.; Si, C.; and Zhang, M.-l. 2023. Complementary Classifier Induced Partial Label Learning. *arXiv preprint arXiv:2305.09897*.

Jin, R.; and Ghahramani, Z. 2002a. Learning with multiple labels. *Advances in neural information processing systems*, 15.

Jin, R.; and Ghahramani, Z. 2002b. Learning with Multiple Labels. In Becker, S.; Thrun, S.; and Obermayer, K., eds., *Advances in Neural Information Processing Systems 15 [Neural Information Processing Systems, NIPS 2002, December 9-14, 2002, Vancouver, British Columbia, Canada]*, 897–904. MIT Press.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Li, Y.; Guo, L.; and Zhou, Z. 2021. Towards Safe Weakly Supervised Learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 43(1): 334–346.

Li, Y.; and Liang, D. 2019. Safe semi-supervised learning: a brief introduction. *Frontiers Comput. Sci.*, 13(4): 669–676.

Liu, L.; and Dietterich, T. G. 2012. A Conditional Multinomial Mixture Model for Superset Label Learning. 557–565.

Liu, L.; and Dietterich, T. G. 2014. Learnability of the Superset Label Learning Problem. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, 1629–1637. JMLR.org.

Lv, J.; Xu, M.; Feng, L.; Niu, G.; Geng, X.; and Sugiyama, M. 2020. Progressive identification of true labels for partial-label learning. In *international conference on machine learning*, 6500–6510. PMLR.

Lyu, G.; Wu, Y.; and Feng, S. 2022. Deep graph matching for partial label learning. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 3306–3312.

Nguyen, N.; and Caruana, R. 2008. Classification with partial labels. In Li, Y.; Liu, B.; and Sarawagi, S., eds., *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, Nevada, USA, August 24-27, 2008*, 551–559. ACM.

Panis, G.; Lanitis, A.; Tsapatsoulis, N.; and Cootes, T. F. 2016. Overview of research on facial ageing using the FG-NET ageing database. *IET Biom.*, 5(2): 37–46.

Papandreou, G.; Chen, L.; Murphy, K. P.; and Yuille, A. L. 2015. Weakly-and Semi-Supervised Learning of a Deep Convolutional Network for Semantic Image Segmentation. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, 1742–1750. IEEE Computer Society.

Qian, W.; Li, Y.; Ye, Q.; Ding, W.; and Shu, W. 2023. Disambiguation-based partial label feature selection via feature dependency and label consistency. *Information Fusion*, 94: 152–168.

Ren, M.; Zeng, W.; Yang, B.; and Urtasun, R. 2018. Learning to Reweight Examples for Robust Deep Learning. In Dy, J. G.; and Krause, A., eds., *Proceedings of the 35th International Conference on Machine Learning, ICML 2018, Stockholmsmässan, Stockholm, Sweden, July 10-15, 2018*, volume 80 of *Proceedings of Machine Learning Research*, 4331–4340. PMLR.

Wang, D.; Zhang, M.; and Li, L. 2022. Adaptive Graph Guided Disambiguation for Partial Label Learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(12): 8796–8811.

Wang, H.; Xiao, R.; Li, Y.; Feng, L.; Niu, G.; Chen, G.; and Zhao, J. 2022. Pico: Contrastive label disambiguation for partial label learning. *arXiv preprint arXiv:2201.08984*.

Wu, D.-D.; Wang, D.-B.; and Zhang, M.-L. 2022. Revisiting consistency regularization for deep partial label learning. In *International Conference on Machine Learning*, 24212–24225. PMLR.

Xia, S.; Lv, J.; Xu, N.; Niu, G.; and Geng, X. 2023. Towards Effective Visual Representations for Partial-Label Learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 15589–15598.

Xu, N.; Lv, J.; and Geng, X. 2019. Partial label learning via label enhancement. In *Proceedings of the AAAI Conference on artificial intelligence*, volume 33, 5557–5564.

Xu, N.; Qiao, C.; Geng, X.; and Zhang, M.-L. 2021. Instance-dependent partial label learning. *Advances in Neural Information Processing Systems*, 34: 27119–27130.

Yu, F.; and Zhang, M. 2017. Maximum margin partial label learning. volume 106, 573–593.

Zeng, Z.; Xiao, S.; Jia, K.; Chan, T.; Gao, S.; Xu, D.; and Ma, Y. 2013. Learning by Associating Ambiguously Labeled Images. In *2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, June 23-28, 2013*, 708–715. IEEE Computer Society.

Zhang, M.; and Yu, F. 2015. Solving the Partial Label Learning Problem: An Instance-Based Approach. 4048–4054.

Zhang, M.; Zhou, B.; and Liu, X. 2016. Partial Label Learning via Feature-Aware Disambiguation. In Krishnapuram, B.; Shah, M.; Smola, A. J.; Aggarwal, C. C.; Shen, D.; and Rastogi, R., eds., *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, 1335–1344. ACM.

Zhang, M.-L.; Wu, J.-H.; and Bao, W.-X. 2022. Disambiguation enabled linear discriminant analysis for partial label dimensionality reduction. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 16(4): 1–18.

Zhou, Z.-H. 2018. A brief introduction to weakly supervised learning. *National science review*, 5(1): 44–53.

Zhu, X.; and Goldberg, A. B. 2009. Introduction to Semi-Supervised Learning.