Provably Consistent Partial-Label Learning

Lei Feng[1]* **Jiaqi Lv**[2] **Bo Han**[3] **Miao Xu**[4,5]
Gang Niu[5] **Xin Geng**[2] **Bo An**[1]† **Masashi Sugiyama**[5,6]
[1]School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2]School of Computer Science and Engineering, Southeast University, Nanjing, China
[3]Department of Computer Science, Hong Kong Baptist University, China
[4]The University of Queensland, Australia
[5]Center for Advanced Intelligence Project, RIKEN, Japan
[6]The University of Tokyo, Japan

Abstract

Partial-label learning (PLL) is a multi-class classification problem, where each training example is associated with *a set of candidate labels*. Even though many practical PLL methods have been proposed in the last two decades, there lacks a theoretical understanding of the consistency of those methods—none of the PLL methods hitherto possesses a *generation process* of candidate label sets, and then it is still unclear why such a method works on a specific dataset and when it may fail given a different dataset. In this paper, we propose the first *generation model* of candidate label sets, and develop two novel PLL methods that are guaranteed to be provably consistent, i.e., one is *risk-consistent* and the other is *classifier-consistent*. Our methods are advantageous, since they are compatible with any deep network or stochastic optimizer. Furthermore, thanks to the generation model, we would be able to answer the two questions above by testing if the generation model matches given candidate label sets. Experiments on benchmark and real-world datasets validate the effectiveness of the proposed generation model and two PLL methods.

1 Introduction

Unlike supervised learning and unsupervised learning, weakly supervised learning [52] aims to learn with weak supervision. So far, various weakly supervised learning frameworks have been widely studied. Examples include *semi-supervised learning* [5, 42, 32], *multi-instance learning* [1, 53], *positive-unlabeled learning* [10, 29], *complementary-label learning* [24, 25], *noisy-label learning* [40, 20, 44], *positive-confidence learning* [26], *similar-unlabeled learning* [2], and *unlabeled-unlabeled learning* [35, 36].

In recent years, another weakly supervised learning framework called *partial-label learning* (PLL) [27, 9, 33, 7, 50, 13, 39] has gradually attracted attention from machine learning and data mining communities. PLL aims to deal with the problem where each instance is provided with a set of candidate labels, only one of which is the correct label. In some studies, PLL is also termed as *ambiguous-label learning* [23, 49, 7, 6, 47] and *superset-label learning* [34, 33, 16]. Due to the difficulty in collecting accurately labeled data in many real-world scenarios, PLL has been successfully applied to a wide range of application domains, such as web mining [37], bird song classification [34], and automatic face naming [49].

A number of methods [27, 41, 50, 12, 13] have been proposed to improve the practical performance of PLL. On the theoretical side, some researchers have studied the statistical consistency [9] and

*Preliminary work was done during an internship at RIKEN AIP.
†Correspondence to: <boan@ntu.edu.sg>.

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

learnability [33] of PLL. They made the same assumption on the *ambiguity degree*, which describes the maximum co-occurring probability of the correct label with another false positive label. Although they assumed that the data distribution for successful PLL should ensure a limited ambiguity degree, it is still unclear what the explicit formulation of the data distribution would be. Besides, the consistency of PLL methods would be hardly guaranteed without modeling the data distribution.

Motivated by the above observations, we for the first time present a novel statistical model to depict the generation process of partially labeled data. Having an explicit data distribution not only helps us to understand how partially labeled examples are generated, but also enables us to perform effective empirical risk minimization. Our proposed data generation model is instance-independent, which does not introduce any extra hidden variable. We verify that the proposed generation model satisfies the key assumption of PLL that the correct label is always included in the set of candidate labels.

Based on the data generation model, we further derive a novel *risk-consistent* method and a novel *classifier-consistent* method. Most of the existing PLL methods need to specially design complex optimization objectives, which make the optimization process inefficient. In contrast, our proposed PLL methods do not rely on specific classification models and can be easily trained with stochastic optimization, thus can be naturally applied to complex models such as deep neural networks with large-scale datasets. In addition, we theoretically derive an estimation error bound for each of the methods, which demonstrates that the obtained empirical risk minimizer would converge to the true risk minimizer as the number of training data tends to infinity. We show that the risk-consistent method holds a tighter estimation error bound than the classifier-consistent method and empirically validate that the risk-consistent method achieves better performance when deep neural networks are used. We also use *entropy* to measure how well the given candidate label sets match our generation model. We find that the candidate label sets with higher entropy better match our generation model, and on such datasets, our proposed PLL methods achieve better performance. Extensive experiments on benchmark as well as real-world partially labeled datasets clearly validate the effectiveness of our proposed methods.

2 Formulations

In this section, we introduce some notations and briefly review the formulations of learning with ordinary labels, learning with partial labels, and learning with complementary labels.

Learning with Ordinary Labels. For ordinary multi-class learning, let the feature space be $\mathcal{X} \in \mathbb{R}^d$ and the label space be $\mathcal{Y} = [k]$ (with k classes) where $[k] := \{1, 2, \ldots, k\}$. Let us clearly define that $\boldsymbol{x}$ denotes an instance and $(\boldsymbol{x}, y)$ denotes an example including an instance $\boldsymbol{x}$ and a label y. When ordinary labels are provided, we usually assume each example $(\boldsymbol{x}, y) \in \mathcal{X} \times \mathcal{Y}$ is independently sampled from an unknown data distribution with probability density $p(\boldsymbol{x}, y)$. Then, the goal of multi-class learning is to obtain a multi-class classifier $f : \mathcal{X} \to \mathbb{R}^k$ that minimizes the following classification risk:

$$R(f) = \mathbb{E}_{p(\boldsymbol{x},y)}[\mathcal{L}(f(\boldsymbol{x}), y)], \tag{1}$$

where $\mathbb{E}_{p(\boldsymbol{x},y)}[\cdot]$ denotes the expectation over the joint probability density $p(\boldsymbol{x}, y)$ and $\mathcal{L} : \mathbb{R}^k \times \mathcal{Y} \to \mathbb{R}_+$ is a multi-class loss function that measures how well a classifier estimates a given label. We say that a method is *classifier-consistent* if the learned classifier by the method is infinite-sample consistent to $\arg\min_{f \in \mathcal{F}} R(f)$, and a method is *risk-consistent* if the method possesses a classification risk estimator that is equivalent to $R(f)$ given the same classifier f. It is worth noting that a risk-consistent method is also classifier-consistent [45]. However, a classifier-consistent method may not be risk-consistent.

Learning with Partial Labels. For learning with partial labels (i.e., PLL), each instance is provided with a set of candidate (partial) labels, only one of which is correct. Suppose the partially labeled dataset is denoted by $\widetilde{\mathcal{D}} = \{(\boldsymbol{x}_i, Y_i)\}_{i=1}^n$ where Y_i is the candidate label set of $\boldsymbol{x}_i$. Since each candidate label set should not be the empty set nor the whole label set, we have $Y_i \in \mathcal{C}$ where $\mathcal{C} = \{2^{\mathcal{Y}} \setminus \emptyset \setminus \mathcal{Y}\}$, $2^{\mathcal{Y}}$ denotes the power set, and $|\mathcal{C}| = 2^k - 2$. The key assumption of PLL lies in that the correct label y_i of $\boldsymbol{x}_i$ must be in the candidate label set, i.e.,

$$p(y_i \in Y_i \mid \boldsymbol{x}_i, Y_i) = 1, \ \forall (\boldsymbol{x}_i, y_i) \in \mathcal{X} \times \mathcal{Y}, \ \forall Y_i \in \mathcal{C}. \tag{2}$$

Given such data, the goal of PLL is to induce a multi-class classifier $f : \mathcal{X} \to \mathbb{R}^k$ that can make correct predictions on test inputs. To this end, many methods [34, 50, 51, 16, 13, 39] have been proposed to improve the performance of PLL. However, to the best of our knowledge, there is only one method [9] that possesses statistical consistency by providing a classifier-consistent risk estimator. However, it not only requires the assumption that the data distribution should ensure a limited ambiguity degree, but also relies on some strict conditions (e.g., convexity of loss function and dominance relation [9]). It is still unclear what the explicit formulation of the data distribution for successful PLL would be. Besides, it is also unknown whether there exists a risk-consistent method that possesses a statistical unbiased estimator of the classification risk $R(f)$.

Learning with Complementary Labels. There is a special case of partial labels, called complementary labels [24, 48, 25]. Each complementary label specifies one of the classes that the example does *not* belong to. Hence a complementary label $\overline{y}$ can be considered as an extreme case where all $k-1$ classes other than the class $\overline{y}$ are taken as candidate (partial) labels. Existing studies on learning with complementary labels make the assumption on the data generation process. The pioneering study [24] assumed that each complementarily labeled example $(\boldsymbol{x}, \overline{y})$ is independently drawn from the probability distribution with density $\overline{p}(\boldsymbol{x}, y)$, where $\overline{p}(\boldsymbol{x}, y)$ is defined as $\overline{p}(\boldsymbol{x}, \overline{y}) = \sum_{y \neq \overline{y}} p(\boldsymbol{x}, y)$. Based on this data distribution, several risk-consistent methods [24, 25] have been proposed for learning with complementary labels. However, in many real-world scenarios, multiple complementary labels would be more widespread than a single complementary label. Hence a recent study [14] focused on learning with multiple complementary labels. Suppose each training example is represented by $(\boldsymbol{x}, \overline{Y})$ where $\overline{Y}$ denotes a set of multiple complementary labels, and $(\boldsymbol{x}, \overline{Y})$ is assumed to be independently sampled from the probability distribution with density $\overline{p}(\boldsymbol{x}, \overline{Y})$, which is defined as

$$\overline{p}(\boldsymbol{x}, \overline{Y}) = \textstyle\sum_{j=1}^{k-1} p(s=j)\overline{p}(\boldsymbol{x}, \overline{Y} \mid s=j), \tag{3}$$

where

$$\overline{p}(\boldsymbol{x}, \overline{Y} \mid s=j) := \begin{cases} \frac{1}{\binom{k-1}{j}} \sum_{y \notin \overline{Y}} p(\boldsymbol{x}, y) & \text{if } |\overline{Y}| = j, \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Here, the variable s denotes the size of the complementary label set. Supplied with this data distribution, a risk-consistent method [14] was proposed. It is worth noting that following the distribution of complementarily labeled data, although we can obtain partial labels by regarding all the complementary labels as non-candidate labels, the resulting distribution of partially labeled data is not explicitly formulated. It would be natural to ask whether there also exists an explicit formulation of the partially labeled data distribution that enables us to derive a novel classifier-consistent method or a novel risk-consistent method that possesses statistical consistency. In this paper, we will give an affirmative answer to this question. Specifically, we will show that based on our proposed data generation model, a novel risk-consistent method (the first one for PLL) and a novel classifier-consistent method can be derived accordingly.

3 Data Generation Model

3.1 Partially Labeled Data Distribution

We assume each partially labeled example $(\boldsymbol{x}, Y)$ is independently drawn from a probability distribution with the following density:

$$\widetilde{p}(\boldsymbol{x}, Y) = \textstyle\sum_{i=1}^{k} p(Y \mid y=i) p(\boldsymbol{x}, y=i), \text{ where } p(Y \mid y=i) = \begin{cases} \frac{1}{2^{k-1}-1} & \text{if } i \in Y, \\ 0 & \text{if } i \notin Y. \end{cases} \tag{5}$$

In Eq. (5), we assume $p(Y \mid \boldsymbol{x}, y) = p(Y \mid y)$, which means, given the correct label y, the candidate label set Y is independent of the instance $\boldsymbol{x}$. This assumption is similar to the conventional modeling of label noise [19] where the observed noisy label is independent of the instance, given the correct label. In addition, there are in total $2^{k-1} - 1$ possible candidate label sets that contain a specific label y. Hence, Eq. (5) describes the probability of each candidate label set being uniformly sampled, given a specific label. Here, we show that our assumed data distribution is a valid probability distribution by the following theorem.

Theorem 1. *The equality* $\int_{\mathcal{C}} \int_{\mathcal{X}} \widetilde{p}(\boldsymbol{x}, Y) \mathrm{d}\boldsymbol{x}\, \mathrm{d}Y = 1$ *holds.*

The proof is provided in Appendix A.1. Given the assumed data distribution in Eq. (5), it would be natural to ask whether our assumed data distribution meets the key assumption of PLL described in Eq. (2), i.e., whether the correct label y is always in the candidate label set Y for every partially labeled example $(\boldsymbol{x}, Y)$ sampled from $\widetilde{p}(\boldsymbol{x}, Y)$. The following theorem provides an affirmative answer to this question.

Theorem 2. *For any partially labeled example* $(\boldsymbol{x}, Y)$ *independently sampled from the assumed data distribution in Eq. (5), the correct label* y *is always in the candidate label set* Y*, i.e.,* $p(y \in Y \mid \boldsymbol{x}, Y) = 1,\ \forall (\boldsymbol{x}, Y) \sim \widetilde{p}(\boldsymbol{x}, Y)$.

The proof is provided in Appendix A.2. Theorem 2 clearly demonstrates that our assumed data distribution in Eq. (5) satisfies the key assumption of PLL.

3.2 Motivation

Here, we provide a motivation why we derived the above data generation model. Generally, a large number of high-quality samples are notably helpful to machine learning or data mining. However, it is usually difficult for our labelers to directly identify the correct label for each instance [52]. Nonetheless, it would be easier to collect a set of candidate labels that contains the correct label. Suppose there is a labeling system that can uniformly sample a label set Y from $\mathcal{C}$. For each instance $\boldsymbol{x}$, the labeling system uniformly samples a label set Y and asks a labeler whether the correct label y is in the sampled label set Y. In this case, the collected examples whose correct label y is included in the proposed label set Y follow the same distribution as Eq. (5). In order to justify that, we first introduce the following lemma.

Lemma 1. *Given any instance* $\boldsymbol{x}$ *with its correct label* y*, for any unknown label set* Y *that is uniformly sampled from* $\mathcal{C}$*, the equality* $p(y \in Y \mid \boldsymbol{x}) = 1/2$ *holds.*

It is quite intuitive to verify that Lemma 1 indeed holds. Specifically, if we do not have any information of Y, we may randomly guess with even probabilities whether the correct y is included in an unknown label set Y or not. A rigorous mathematical proof is provided in Appendix A.3. Based on Lemma 1, we have the following theorem.

Theorem 3. *In the above setting, the distribution of the collected data whose correct label* $y \in \mathcal{Y}$ *is included in the label set* $Y \in \mathcal{C}$ *is the same as Eq. (5), i.e.,* $p(\boldsymbol{x}, Y \mid y \in Y) = \widetilde{p}(\boldsymbol{x}, Y)$ *where* $\widetilde{p}(\boldsymbol{x}, Y)$ *is defined in Eq. (5).*

The proof is provided in Appendix A.4.

4 Consistent Methods

In this section, based on our assumed partially labeled data distribution in Eq. (5), we present a novel risk-consistent method and a novel classifier-consistent method and theoretically derive an estimator error bound for each of them. Both methods are agnostic in specific classification models and can be easily trained with stochastic optimization, which ensures their scalability to large-scale datasets.

4.1 Risk-Consistent Method

For the risk-consistent method, we employ the *importance reweighting* strategy [17] to rewrite the classification risk $R(f)$ as

$$
\begin{aligned}
R(f) &= \mathbb{E}_{p(\boldsymbol{x},y)}[\mathcal{L}\big(f(\boldsymbol{x}), y\big)] = \int_{\boldsymbol{x}} \sum_{i=1}^{k} p(y=i \mid \boldsymbol{x}) \mathcal{L}\big(f(\boldsymbol{x}), i\big) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \int_{\boldsymbol{x}} \sum_{i=1}^{k} \frac{1}{|\mathcal{C}|} \sum_{Y \in \mathcal{C}} p(Y \mid \boldsymbol{x}) \frac{p(y=i|\boldsymbol{x})}{p(Y|\boldsymbol{x})} \mathcal{L}\big(f(\boldsymbol{x}), i\big) p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \frac{1}{|\mathcal{C}|} \int_{\boldsymbol{x}} \sum_{Y \in \mathcal{C}} p(Y|\boldsymbol{x}) \Big[\sum_{i=1}^{k} \frac{p(y=i|\boldsymbol{x})}{p(Y|\boldsymbol{x})} \mathcal{L}(f(\boldsymbol{x}), i) \Big] p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} \\
&= \frac{1}{2^k - 2} \mathbb{E}_{\widetilde{p}(\boldsymbol{x},Y)} \Big[\sum_{i=1}^{k} \frac{p(y=i|\boldsymbol{x})}{p(Y|\boldsymbol{x})} \mathcal{L}\big(f(\boldsymbol{x}), i\big) \Big] = R_{\mathrm{rc}}(f). \qquad (6)
\end{aligned}
$$

Here, $p(Y \mid \boldsymbol{x})$ can be calculated by

$$p(Y \mid \boldsymbol{x}) = \sum_{j=1}^{k} p(Y \mid y=j) p(y=j \mid \boldsymbol{x}) = \frac{1}{2^{k-1}-1} \sum_{j \in Y} p(y=j \mid \boldsymbol{x}), \tag{7}$$

where the last equality holds due to Eq. (5). By substituting Eq. (7) into Eq. (6), we obtain

$$R_{\mathrm{rc}}(f) = \frac{1}{2} \mathbb{E}_{\widetilde{p}(\boldsymbol{x},Y)} \Big[\sum_{i=1}^{k} \frac{p(y=i|\boldsymbol{x})}{\sum_{j \in Y} p(y=j|\boldsymbol{x})} \mathcal{L}\big(f(\boldsymbol{x}), i\big) \Big]. \tag{8}$$

In this way, its empirical risk estimator can be expressed as

$$\widehat{R}_{\mathrm{rc}}(f) = \frac{1}{2n} \sum_{o=1}^{n} \Big(\sum_{i=1}^{k} \frac{p(y_o=i|\boldsymbol{x}_o)}{\sum_{j \in Y_o} p(y_o=j|\boldsymbol{x}_o)} \mathcal{L}\big(f(\boldsymbol{x}_o), i\big) \Big), \tag{9}$$

where $\{\boldsymbol{x}_o, Y_o\}_{o=1}^{n}$ are partially labeled examples drawn from $\widetilde{p}(\boldsymbol{x}, Y)$. Note that $p(y=i \mid \boldsymbol{x})$ is not accessible from the given data. Therefore, we apply the softmax function on the model output $f(\boldsymbol{x})$ to approximate $p(y=i \mid \boldsymbol{x})$, i.e., $p(y=i \mid \boldsymbol{x}) = g_i(\boldsymbol{x})$ where $g_i(\boldsymbol{x})$ is the probability of label i being the true label of $\boldsymbol{x}$, which is calculated by $g_i(\boldsymbol{x}) = \exp(f_i(\boldsymbol{x})) / \sum_{j=1}^{k} \exp(f_j(\boldsymbol{x}))$, and $f_i(\boldsymbol{x})$ is the i-th coordinate of $f(\boldsymbol{x})$. Note that the non-candidate labels can never be the correct label. Hence we further correct $p(y=i \mid \boldsymbol{x})$ by setting the confidence of each non-candidate label to 0, i.e.,

$$p(y=i \mid \boldsymbol{x}) = g_i(\boldsymbol{x}) \text{ if } i \in Y, \text{ otherwise } p(y=i \mid \boldsymbol{x}) = 0, \ \forall (\boldsymbol{x}, Y) \sim \widetilde{p}(\boldsymbol{x}, Y). \tag{10}$$

As shown in Eq. (9), our risk-consistent method does not rely on specific loss functions, hence we simply adopt the widely-used categorical cross entropy loss for practical implementation. The pseudo-code of the Risk-Consistent (RC) method is presented in Algorithm 1. It is worth noting that the algorithmic process of RC surprisingly coincides with that of PRODEN [38]. However, they are derived in totally different manners. Besides, PRODEN does not hold any theoretical guarantee while we show that our proposed RC method is consistent.

Here, we establish an estimation error bound for our RC method to demonstrate its learning consistency. Let $\widehat{f}_{\mathrm{rc}} = \min_{f \in \mathcal{F}} \widehat{R}_{\mathrm{rc}}(f)$ be the empirical risk minimizer and $f^{\star} = \min_{f \in \mathcal{F}} R(f)$ be the true risk minimizer. Besides, we define the function space $\mathcal{H}_y$ for the label $y \in \mathcal{Y}$ as $\{h : \boldsymbol{x} \mapsto f_y(\boldsymbol{x}) \mid f \in \mathcal{F}\}$. Let $\mathfrak{R}_n(\mathcal{H}_y)$ be the expected Rademacher complexity [3] of $\mathcal{H}_y$ with sample size n, then we have the following theorem.

Theorem 4. *Assume the loss function $\mathcal{L}(f(\boldsymbol{x}), y)$ is ρ-Lipschitz with respect to $f(\boldsymbol{x})$ ($0 < \rho < \infty$) for all $y \in \mathcal{Y}$ and upper-bounded by M, i.e., $M = \sup_{\boldsymbol{x} \in \mathcal{X}, f \in \mathcal{F}, y \in \mathcal{Y}} \mathcal{L}(f(\boldsymbol{x}), y)$. Then, for any $\delta > 0$, with probability at least $1-\delta$,*

$$R(\widehat{f}_{\mathrm{rc}}) - R(f^{\star}) \leq 4\sqrt{2}\rho \sum_{y=1}^{k} \mathfrak{R}_n(\mathcal{H}_y) + M \sqrt{\frac{\log \frac{2}{\delta}}{2n}},$$

The proof of Theorem 4 is provided in Appendix B. Generally, $\mathfrak{R}_n(\mathcal{H}_y)$ can be bounded by $C_{\mathcal{H}}/\sqrt{n}$ for a positive constant $C_{\mathcal{H}}$ [36, 45, 15]. Hence Theorem 4 shows that the empirical risk minimizer f_{rc} converges to the true risk minimizer $f^{\star}$ as $n \to \infty$.

4.2 Classifier-Consistent Method

For the classifier-consistent method, we start by introducing a transition matrix $\boldsymbol{Q}$ that describes the probability of the candidate label set given an ordinary label. Specifically, the transition matrix $\boldsymbol{Q}$ is defined as $Q_{ij} = p(Y = C_j \mid y = i)$ where $C_j \in \mathcal{C}$ ($j \in [2^k - 2]$) is a specific label set. By further taking into account the assumed data distribution in Eq. (5), we can instantiate the transition matrix $\boldsymbol{Q}$ as $Q_{ij} = \frac{1}{2^{k-1}-1}$ if $i \in C_j$, otherwise $Q_{ij} = 0$. Let us introduce $q_j(\boldsymbol{x}) = p(Y = C_j \mid \boldsymbol{x})$ and $g_i(\boldsymbol{x}) = p(y = i \mid \boldsymbol{x})$, then we can obtain $q(\boldsymbol{x}) = \boldsymbol{Q}^{\top} g(\boldsymbol{x})$ with the assumption $p(Y \mid \boldsymbol{x}, y) = p(Y \mid y)$. Given each partially labeled example $(\boldsymbol{x}, Y)$ sampled from $\widetilde{p}(\boldsymbol{x}, Y)$, the proposed classifier-consistent risk estimator is presented as

$$R_{\mathrm{cc}}(f) = \mathbb{E}_{\widetilde{p}(\boldsymbol{x},Y)}[\mathcal{L}(q(\boldsymbol{x}), \widetilde{y})], \text{ where } Y = C_{\widetilde{y}}. \tag{11}$$

In this formulation, we regard the candidate label set Y as a virtual label $\widetilde{y}$ if Y is a specific label set $C_{\widetilde{y}}$. Since there are $2^k - 2$ possible label sets, we denote by $\widetilde{\mathcal{Y}}$ the virtual label space where $\widetilde{\mathcal{Y}} = [2^k - 2]$ and $\widetilde{y} \in \widetilde{\mathcal{Y}}$. It is worth noting that the transition matrix $\boldsymbol{Q}$ has full rank, because all rows of $\boldsymbol{Q}$ are linearly independent by the definition of $\boldsymbol{Q}$. Then, in order to prove that this method is classifier-consistent, we introduce the following lemma.

Lemma 2. *If certain loss functions are used (e.g., the softmax cross entropy loss or mean squared error), by minimizing the expected risk $R(f)$, the optimal mapping $g^{\star}$ satisfies $g_i^{\star}(\boldsymbol{x}) = p(y = i \mid \boldsymbol{x})$.*

The proof is provided in Appendix C.1. The same proof can also be found in [48, 38].

Algorithm 1 RC Algorithm

Input: Model f, epoch $T_{\max}$, iteration $I_{\max}$, partially labeled training set $\widetilde{\mathcal{D}} = \{(\boldsymbol{x}_i, Y_i)\}_{i=1}^{n}$.

Initialize $p(y_i = j \mid \boldsymbol{x}_i) = 1, \forall j \in Y_i$, otherwise $p(y_i = j \mid \boldsymbol{x}_i) = 0$;
for $t = 1, 2, \ldots, T_{\max}$ **do**
  Shuffle $\widetilde{\mathcal{D}} = \{(\boldsymbol{x}_i, Y_i)\}_{i=1}^{n}$;
  for $j = 1, \ldots, I_{\max}$ **do**
    Fetch mini-batch $\widetilde{\mathcal{D}}_j$ from $\widetilde{\mathcal{D}}$;
    Update model f by $\widehat{R}_{\mathrm{rc}}$ in Eq. (9);
    Update $p(y_i \mid \boldsymbol{x}_i)$ by Eq. (10);
  end for
end for **Output:** f.

Algorithm 2 CC Algorithm

Input: Model f, epoch $T_{\max}$, iteration $I_{\max}$, partially labeled training set $\widetilde{\mathcal{D}} = \{(\boldsymbol{x}_i, Y_i)\}_{i=1}^{n}$;

for $t = 1, 2, \ldots, T_{\max}$ **do**
  Shuffle the partially labeled training set $\widetilde{\mathcal{D}} = \{(\boldsymbol{x}_i, Y_i)\}_{i=1}^{n}$;
  for $j = 1, \ldots, I_{\max}$ **do**
    Fetch mini-batch $\widetilde{\mathcal{D}}_j$ from $\widetilde{\mathcal{D}}$;
    Update model f by minimizing the empirical risk estimator $\widehat{R}_{\mathrm{cc}}$ in Eq. (12);
  end for
end for

Output: f.

Theorem 5. *When the transition matrix $\boldsymbol{Q}$ has full rank and the condition in Lemma 2 is satisfied, the minimizer $f_{\mathrm{cc}} = \arg\min_{f \in \mathcal{F}} R_{\mathrm{cc}}(f)$ is also the true minimizer $f^\star = \arg\min_{f \in \mathcal{F}} R(f)$, i.e., $f_{\mathrm{cc}} = f^\star$ (classifier-consistency).*

The proof is provided in Appendix C.2.

As suggested by Lemma 2, we adopt the cross entropy loss in our classifier-consistent risk estimator (i.e., Eq. (11)) for practical implementation. In this way, we have the following empirical risk estimator:

$$\widehat{R}_{\mathrm{cc}}(f) = -\frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{2^k-2}\mathbb{I}(Y_i = C_j)\log(q_j(\boldsymbol{x}_i))\right) = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{2^k-2}\mathbb{I}(Y_i = C_j)\log\left(\boldsymbol{Q}[:, j]^\top g(\boldsymbol{x})\right)$$
$$= -\frac{1}{n}\sum_{i=1}^{n}\log\left(\frac{1}{2^{k-1}-1}\sum_{y \in Y_i} g_y(\boldsymbol{x})\right) = -\frac{1}{n}\sum_{i=1}^{n}\log\left(\frac{1}{2^{k-1}-1}\sum_{y \in Y_i}\frac{\exp(f_y(\boldsymbol{x}))}{\sum_j \exp(f_j(\boldsymbol{x}))}\right), \tag{12}$$

where $\mathbb{I}[\cdot]$ is the indicator function. For the expected risk estimator $R_{\mathrm{cc}}(f)$, it seems that the transition matrix $\boldsymbol{Q} \in \mathbb{R}^{k \times (2^k - 2)}$ is indispensable. Unfortunately, it would be computationally prohibitive, since $2^k - 2$ is an extremely large number if the number of classes k is large. However, for practical implementation, Eq. (12) shows that we do not need to explicitly calculate and store the transition matrix $\boldsymbol{Q}$, which brings no pain to optimization. The pseudo-code of the Classifier-Consistent (CC) method is presented in Algorithm 2.

Here, we also establish an estimation error bound for the classifier-consistent method. Let $\widehat{f}_{\mathrm{cc}} = \arg\min_{f \in \mathcal{F}} \widehat{R}_{\mathrm{cc}}(f)$ be the empirical minimizer and $f^\star = \arg\min_{f \in \mathcal{F}} R(f)$ be the true minimizer. Besides, we define the function space $\mathcal{H}_y$ for the label $y \in \mathcal{Y}$ as $\{h : \boldsymbol{x} \mapsto f_y(\boldsymbol{x}) \mid f \in \mathcal{F}\}$. Then, we have the following theorem.

Theorem 6. *Assume the loss function $\mathcal{L}(q(\boldsymbol{x}), \widetilde{y})$ is ρ'-Lipschitz with respect to $f(\boldsymbol{x})$ $(0 < \rho < \infty)$ for all $\widetilde{y} \in \widetilde{\mathcal{Y}}$ and upper-bounded by M, i.e., $M = \sup_{\boldsymbol{x} \in \mathcal{X}, f \in \mathcal{F}, \widetilde{y} \in \widetilde{\mathcal{Y}}} \mathcal{L}(q(\boldsymbol{x}), \widetilde{y})$. Then, for any $\delta > 0$, with probability at least $1 - \delta$,*

$$R_{\mathrm{cc}}(\widehat{f}_{cc}) - R_{\mathrm{cc}}(f^\star) \leq 4\sqrt{2}\rho' \textstyle\sum_{y=1}^{k} \mathfrak{R}_n(\mathcal{H}_y) + 2M\sqrt{\frac{\log\frac{2}{\delta}}{2n}}.$$

The proof is provided in Appendix D. Theorem 6 demonstrates that the empirical risk minimizer $\widehat{f}_{\mathrm{cc}}$ converges to the true risk minimizer $f^\star$ as $n \to \infty$.

Theoretical Comparison Between RC and CC. There exists a clear difference between the estimation error bounds in Theorem 4 and Theorem 6, especially in the last term. If we assume that ρ for RC and ρ' for CC hold the same value, we can find that the estimation error bound in Theorem 6 would be looser than that in Theorem 4. Therefore, we could expect that RC may have better performance than CC. In addition, RC needs to estimate the prediction confidence of each example. Intuitively, complex models like deep neural networks normally provide more accurate estimation than linear models. Therefore, we speculate that when more complex models are used, the superiority of RC would be more remarkable. We will demonstrate via experiments that RC is generally superior to CC when deep neural networks are used.

Table 1: Test performance (mean±std) of each method using neural networks on benchmark datasets. ResNet is trained on CIFAR-10, and MLP is trained on the other three datasets.

	MNIST	Kuzushiji-MNIST	Fashion-MNIST	CIFAR-10
RC	**98.00±0.11%**	**89.38±0.28%**	**88.38±0.16%**	**77.93±0.59%**
CC	97.87±0.10%•	88.83±0.40%•	87.88±0.25%•	75.78±0.27%•
GA	96.37±0.13%•	84.23±0.19%•	85.57±0.16%•	72.22±0.19%•
NN	96.75±0.08%•	82.36±0.41%•	86.25±0.14%•	68.09±0.31%•
Free	88.48±0.37%•	70.31±0.68%•	81.34±0.47%•	17.74±1.20%•
PC	92.47±0.13%•	73.45±0.20%•	83.37±0.31%•	46.53±2.01%•
Forward	97.64±0.11%•	87.64±0.13%•	86.73±0.15%•	71.18±0.92%•
EXP	97.81±0.04%•	88.48±0.29%•	87.96±0.06%•	73.22±0.66%•
LOG	97.86±0.11%•	88.24±0.08%•	88.31±0.26%	75.38±0.34%•
MAE	97.82±0.11%•	88.43±0.32%•	87.83±0.22%•	66.91±3.08%•
MSE	96.95±0.14%•	85.16±0.44%•	85.72±0.26%•	66.15±2.13%•
GCE	96.71±0.08%•	85.19±0.39%•	86.88±0.16%•	68.39±0.71%•
Phuber-CE	95.10±0.34%•	80.66±0.41%•	85.33±0.23%•	58.60±0.95%•

Table 2: Test performance (mean±std) of each method using neural networks on benchmark datasets. DenseNet is trained on CIFAR-10, and LeNet is trained on the other three datasets.

	MNIST	Kuzushiji-MNIST	Fashion-MNIST	CIFAR-10
RC	**99.04±0.03%**	**94.00±0.30%**	**89.48±0.15%**	**78.53±0.46%**
CC	98.99±0.08%	93.86±0.18%	88.98±0.20%•	75.71±0.18%•
GA	98.68±0.05%•	90.39±0.26%•	87.95±0.12%•	71.85±0.19%•
NN	98.51±0.08%•	89.60±0.34%•	88.47±0.15%•	71.98±0.35%•
Free	80.48±2.06%•	71.18±1.38%•	74.02±3.88%•	45.94±0.83%•
PC	95.03±0.16%•	79.62±0.11%•	83.98±0.20%•	54.18±2.10%•
Forward	98.80±0.04%•	93.87±0.14%	88.72±0.17%•	73.56±1.47%•
EXP	98.82±0.03%•	92.69±0.31%•	88.99±0.25%•	75.02±1.02%•
LOG	98.88±0.08%•	93.97±0.25%	88.75±0.28%•	75.54±0.59%•
MAE	98.88±0.05%•	93.04±0.52%•	87.30±3.16%•	67.74±0.89%•
MSE	98.38±0.05%•	88.37±0.55%•	88.18±0.08%•	70.66±0.59%•
GCE	98.63±0.06%•	91.27±0.30%•	88.66±0.16%•	72.09±0.51%•
Phuber-CE	96.92±0.18%•	82.24±2.45%•	87.02±0.09%•	66.47±0.35%•

5 Experiments

In this section, we conduct extensive experiments on various datasets to validate the effectiveness of our proposed methods.

Datasets. We collect four widely used benchmark datasets including MNIST [31], Kuzushiji-MNIST [8], Fashion-MNIST [46], and CIFAR-10 [30], and five datasets from the UCI Machine Learning Repository [30]. In order to generate candidate label sets on these datasets, following the motivation in Section 3.2, we uniformly sample the candidate label set that includes the correct label from $\mathcal{C}$ for each instance. In addition, we also use five widely used real-world partially labeled datasets, including Lost [9], BirdSong [4], MSRCv2 [34], Soccer Player [49], Yahoo! News [18]. Since our proposed methods do not rely on specific classification models, we use various base models to validate the effectiveness of our methods, including linear model, three-layer (d-500-k) MLP, 5-layer LeNet, 34-layer ResNet [21], and 22-layer DenseNet [22]. The detailed descriptions of these datasets with the corresponding base models are provided in Appendix E.1.

Compared Methods. We compare with six state-of-the-art PLL methods including SURE [13], CLPL [9], IPAL [50], PLSVM [11], PLECOC [51], PLKNN [23]. Besides, we also compare with various *complementary-label learning* (CLL) methods for two reasons: 1) We can directly use CLL methods on partially labeled datasets by regarding non-candidate labels as complementary labels. 2) Existing CLL methods can be applied to large-scale datasets. The compared CLL methods include GA, NN, and Free [25], PC [24], Forward [48], the unbiased risk estimator [14] with bounded losses MAE, MSE, GCE, Phuber-CE, and the surrogate losses EXP and LOG. For all the above methods, their hyper-parameters are specified or searched according to the suggested parameter settings by respective papers. The detailed information of these compared methods is provided in Appendix E.2. For our proposed methods RC (Algorithm 1) and CC (Algorithm 2), we only need to search learning rate and weight decay from $\{10^{-6}, \ldots, 10^{-1}\}$, since there are no other hyper-parameters in our methods. Hyper-parameters are selected so as to maximize the accuracy on a validation set

Table 3: Test performance (mean±std) of each method using linear model on UCI datasets.

	Texture	Yeast	Dermatology	Har	20Newsgroups
RC	99.24±0.14%	59.89±1.27%	99.41±1.00%	98.03±0.09%	**75.99±0.53%**
CC	98.02±2.91%•	**59.97±1.57%**	**99.73±0.85%**	**98.10±0.18%**	75.97±0.54%
SURE	95.38±0.28%•	54.39±1.32%•	97.48±0.32%•	97.43±0.24%•	69.82±0.26%•
CLPL	91.93±0.97%•	54.58±2.11%•	99.62±0.85%	97.48±0.18%•	71.44±0.55%•
PLECOC	69.69±4.82%•	37.37±9.73%•	87.84±5.30%•	96.97±0.29%•	15.32±7.86%•
PLSVM	49.38±9.99%•	45.70±8.01%•	80.00±7.53%•	91.64±1.43%•	32.59±8.91%•
PLKNN	96.78±0.31%•	47.79±2.41%•	80.54±5.06%•	94.17±0.59%•	27.18±0.65%•
IPAL	**99.45±0.23%**	48.99±3.84%•	98.65±2.27%•	96.55±0.40%•	48.36±0.85%•

Table 4: Test performance (mean±std) of each method using linear model on real-world datasets.

	Lost	MSRCv2	BirdSong	Soccer Player	Yahoo! News
RC	**79.43±3.26%**	46.56±2.71%	71.94±1.72%	**57.00±0.97%**	**68.23±0.83%**
CC	79.29±3.19%	47.22±3.02%	**72.22±1.71%**	56.32±0.64%	68.14±0.81%
SURE	71.33±3.57%•	46.88±4.67%	58.92±1.28%•	49.41±086%•	45.49±1.15%•
CLPL	74.87±4.30%•	36.53±4.59%•	63.56±1.40%•	36.82±1.04%•	46.21±0.90%•
PLECOC	49.03±8.36%•	41.53±3.25%•	71.58±1.81%	53.70±2.02%•	66.22±1.01%•
PLSVM	75.31±3.81%•	35.85±4.41%•	49.90±2.07%•	46.29±0.96%•	56.85±0.91%•
PLKNN	36.73±2.99%•	41.36±2.89%•	64.94±1.42%•	49.62±0.67%•	41.07±1.02%•
IPAL	72.12±4.48%•	**50.80±4.46%**○	72.06±1.55%	55.03±0.77%•	66.79±1.22%•

(10% of the training set) of partially labeled data. We implement them using PyTorch [43] and use the Adam [28] optimizer with the mini-batch size set to 256 and the number of epochs set to 250.

Experimental Results. We run 5 trials on the four benchmark datasets and run 10 trials (with 90%/10% train/test split) on UCI datasets and real-world partially labeled datasets, and record the mean accuracy with standard deviation (mean±std). We also use paired t-test at 5% significance level, and •/○ represents whether the *best* of RC and CC is significantly better/worse than other compared methods. Besides, the best results are highlighted in bold. Table 1 and Table 2 report the test performance of each method using neural networks on benchmark datasets. We also provide the transductive performance of each method in Appendix E.3. From the two tables, we can observe that RC always achieves the best performance and significantly outperforms other compared methods in most cases. In addition, we record the test accuracy at each training epoch to provide more detailed visualized results in Appendix E.4. Table 3 and Table 4 report the test performance of each method using linear model on UCI datasets and real-world partially labeled datasets, respectively. We can find that RC and CC generally achieve superior performance against other compared methods on both UCI datasets and real-world partially labeled datasets.

Performance Comparison Between RC and CC. It can be seen that when linear model is used, RC and CC achieve similar performance. However, RC significantly outperforms CC when deep neural networks are used. These observations clearly accord with our conjecture that the superiority of RC would be more remarkable when more complex models are used.

Effectiveness of Generation Model. We use *entropy* to measure how well given candidate label sets match the proposed generation model. By this measure, we could know ahead of model training whether to apply our proposed methods or not on a specific dataset. We expect that the higher the entropy, the better the match, thus the better the performance of our proposed methods. To verify our conjecture, we generate various candidate labels sets by different generation models, and the experimental results agree with our conjecture. We further show via experiments that even when given candidate label sets do not match our proposed generation model well, our methods still significantly outperform other compared methods. These experimental results are provided in Appendix F.

6 Conclusion

In this paper, we for the first time provided an explicit mathematical formulation of the partially labeled data generation process for PLL. Based on our data generation model, we further derived a novel *risk-consistent* method and a novel *classifier-consistent* method. To the best of our knowledge, we provided the first risk-consistent PLL method. Besides, our proposed methods do not reply on specific models and can be easily trained with stochastic optimization, which ensures their scalability to large-scale datasets. In addition, we theoretically derived an *estimation error bound* for each of the proposed methods. Finally, extensive experimental results clearly demonstrated the effectiveness of the proposed generation model and two PLL methods.

Broader Impact

A potential application of our proposed partial-label learning methods would be data privacy. For example, when we collect some survey data, we may ask respondents to answer some extremely private questions. It would be difficult for us to directly obtain the ground-truth answer (label) to the question. However, it would be easier for us to obtain a set of candidate labels that contains the true label, since it is mentally less demanding for respondents to remove several obviously wrong labels. In this case, our proposed partial-label learning methods can be used.

There may also exist some negative impacts of our proposed methods. For example, an adversary might deliberately ask a person to give some candidate choices or remove some improper choices to specially designed questions, so that high-quality partially labeled data could be collected. The adversary may apply the proposed partial-label learning methods to learn from the collected partially labeled data. As a consequence, some extremely private data of the person would be divulged or leveraged by the adversary. In addition, if partial-label learning methods are very effective and prevalent, the need for accurately annotated data would be significantly reduced. As a result, the rate of unemployment for data annotation specialists might be increased.

Acknowledgements

This research was supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG-RP-2019-0013), National Satellite of Excellence in Trustworthy Software Systems (Award No: NSOE-TSS2019-01), and NTU. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore. JL and XG were supported by NSFC (62076063). BH was supported by the RGC Early Career Scheme No. 22200720, NSFC Young Scientists Fund No. 62006202, HKBU Tier-1 Start-up Grant and HKBU CSD Start-up Grant. GN and MS were supported by JST AIP Acceleration Research Grant Number JPMJCR20U3, Japan, and MS was also supported by Institute for AI and Beyond, UTokyo.

References

[1] J. Amores. Multiple instance classification: Review, taxonomy and comparative study. *Artificial Intelligence*, 201:81–105, 2013.

[2] H. Bao, G. Niu, and M. Sugiyama. Classification from pairwise similarity and unlabeled data. In *ICML*, pages 452–461, 2018.

[3] P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *JMLR*, 3(11):463–482, 2002.

[4] F. Briggs, X. Z. Fern, and R. Raich. Rank-loss support instance machines for miml instance annotation. In *KDD*, pages 534–542, 2012.

[5] O. Chapelle, B. Scholkopf, and A. Zien. *Semi-Supervised Learning*. MIT Press, 2006.

[6] C.-H. Chen, V. M. Patel, and R. Chellappa. Learning from ambiguously labeled face images. *TPAMI*, 40(7):1653–1667, 2018.

[7] Y.-C. Chen, V. M. Patel, R. Chellappa, and P. J. Phillips. Ambiguously labeled learning using dictionaries. *TIFS*, 9(12):2076–2088, 2014.

[8] T. Clanuwat, M. Bober-Irizar, A. Kitamoto, A. Lamb, K. Yamamoto, and D. Ha. Deep learning for classical japanese literature. *arXiv preprint arXiv:1812.01718*, 2018.

[9] T. Cour, B. Sapp, and B. Taskar. Learning from partial labels. *JMLR*, 12(5):1501–1536, 2011.

[10] M. C. du Plessis, G. Niu, and M. Sugiyama. Convex formulation for learning from positive and unlabeled data. In *ICML*, pages 1386–1394, 2015.

[11] C. Elkan and K. Noto. Learning classifiers from only positive and unlabeled data. In *KDD*, pages 213–220, 2008.

[12] L. Feng and B. An. Leveraging latent label distributions for partial label learning. In *IJCAI*, pages 2107–2113, 2018.

[13] L. Feng and B. An. Partial label learning with self-guided retraining. In *AAAI*, pages 3542–3549, 2019.

[14] L. Feng, T. Kaneko, B. Han, G. Niu, B. An, and M. Sugiyama. Learning with multiple complementary labels. In *ICML*, 2020.

[15] N. Golowich, A. Rakhlin, and O. Shamir. Size-independent sample complexity of neural networks. *arXiv preprint arXiv:1712.06541*, 2017.

[16] C. Gong, T.-L. Liu, Y.-Y. Tang, J. Yang, J. Yang, and D.-C. Tao. A regularization approach for instance-based superset label learning. *IEEE Transactions on Cybernetics*, 48(3):967–978, 2018.

[17] A. Gretton, A. Smola, J.-Y. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. *Dataset Shift in Machine Learning*, 3(4):5, 2009.

[18] M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. *Lecture Notes in Computer Science*, 63(11):634–647, 2010.

[19] B. Han, J.-C. Yao, G. Niu, M.-Y. Zhou, I. Tsang, Y. Zhang, and M. Sugiyama. Masking: A new perspective of noisy supervision. In *NeurIPS*, pages 5836–5846, 2018.

[20] B. Han, Q.-M. Yao, X.-R. Yu, G. Niu, M. Xu, W.-H. Hu, I. Tsang, and M. Sugiyama. Co-teaching: Robust training of deep neural networks with extremely noisy labels. In *NeurIPS*, pages 8527–8537, 2018.

[21] K.-M. He, X.-Y. Zhang, S.-Q. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[22] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *CVPR*, pages 4700–4708, 2017.

[23] E. Hüllermeier and J. Beringer. Learning from ambiguously labeled examples. *Intelligent Data Analysis*, 10(5):419–439, 2006.

[24] T. Ishida, G. Niu, W.-H. Hu, and M. Sugiyama. Learning from complementary labels. In *NeurIPS*, pages 5644–5654, 2017.

[25] T. Ishida, G. Niu, A. K. Menon, and M. Sugiyama. Complementary-label learning for arbitrary losses and models. In *ICML*, pages 2971–2980, 2019.

[26] T. Ishida, G. Niu, and M. Sugiyama. Binary classification for positive-confidence data. In *NeurIPS*, pages 5917–5928, 2018.

[27] R. Jin and Z. Ghahramani. Learning with multiple labels. In *NeurIPS*, pages 921–928, 2003.

[28] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[29] R. Kiryo, G. Niu, M. C. du Plessis, and M. Sugiyama. Positive-unlabeled learning with non-negative risk estimator. In *NeurIPS*, pages 1674–1684, 2017.

[30] A. Krizhevsky, G. Hinton, et al. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

[31] Y. LeCun, L. Bottou, Y. Bengio, P. Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[32] Y.-F. Li and D.-M. Liang. Safe semi-supervised learning: a brief introduction. *Frontiers of Computer Science*, 13(4):669–676, 2019.

[33] L.-P. Liu and T. Dietterich. Learnability of the superset label learning problem. In *ICML*, pages 1629–1637, 2014.

[34] L.-P. Liu and T. G. Dietterich. A conditional multinomial mixture model for superset label learning. In *NeurIPS*, pages 548–556, 2012.

[35] N. Lu, G. Niu, A. K. Menon, and M. Sugiyama. On the minimal supervision for training any binary classifier from only unlabeled data. In *ICLR*, 2019.

[36] N. Lu, T.-Y. Zhang, G. Niu, and M. Sugiyama. Mitigating overfitting in supervised classification from two unlabeled datasets: A consistent risk correction approach. In *AISTATS*, 2020.

[37] J. Luo and F. Orabona. Learning from candidate labeling sets. In *NeurIPS*, pages 1504–1512, 2010.

[38] J. Lv, M. Xu, L. Feng, G. Niu, X. Geng, and M. Sugiyama. Progressive identification of true labels for partial-label learning. *arXiv preprint arXiv:2002.08053*, 2020.

[39] G.-Y. Lyu, S.-H. Feng, T. Wang, C.-Y. Lang, and Y.-D. Li. Gm-pll: Graph matching based partial label learning. *TKDE*, 2019.

[40] A. Menon, B. Van Rooyen, C. S. Ong, and B. Williamson. Learning from corrupted binary labels via class-probability estimation. In *ICML*, pages 125–134, 2015.

[41] N. Nguyen and R. Caruana. Classification with partial labels. In *KDD*, pages 551–559, 2008.

[42] G. Niu, W. Jitkrittum, B. Dai, H. Hachiya, and M. Sugiyama. Squared-loss mutual information regularization: A novel information-theoretic approach to semi-supervised learning. In *ICML*, pages 10–18, 2013.

[43] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, pages 8024–8035, 2019.

[44] H.-X. Wei, L. Feng, X.-Y. Chen, and B. An. Combating noisy labels by agreement: A joint training method with co-regularization. In *CVPR*, pages 13726–13735, 2020.

[45] X.-B. Xia, T.-L. Liu, N.-N. Wang, B. Han, C. Gong, G. Niu, and M. Sugiyama. Are anchor points really indispensable in label-noise learning? In *NeurIPS*, pages 6835–6846, 2019.

[46] H. Xiao, K. Rasul, and R. Vollgraf. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

[47] Y. Yao, C. Gong, J.-H. Deng, X.-H. Chen, J.-X. Wu, and J. Yang. Deep discriminative cnn with temporal ensembling for ambiguously-labeled image classification. In *AAAI*, 2020.

[48] X.-Y. Yu, T.-L. Liu, M.-M. Gong, and D.-C. Tao. Learning with biased complementary labels. In *ECCV*, pages 68–83, 2018.

[49] Z.-N. Zeng, S.-J. Xiao, K. Jia, T.-H. Chan, S.-H. Gao, D. Xu, and Y. Ma. Learning by associating ambiguously labeled images. In *CVPR*, pages 708–715, 2013.

[50] M.-L. Zhang and F. Yu. Solving the partial label learning problem: An instance-based approach. In *IJCAI*, pages 4048–4054, 2015.

[51] M.-L. Zhang, F. Yu, and C.-Z. Tang. Disambiguation-free partial label learning. *TKDE*, 29(10):2155–2167, 2017.

[52] Z.-H. Zhou. A brief introduction to weakly supervised learning. *National Science Review*, 5(1):44–53, 2018.

[53] Z.-H. Zhou, M.-L. Zhang, S.-J. Huang, and Y.-F. Li. Multi-instance multi-label learning. *Artificial Intelligence*, 176(1):2291–2320, 2012.